AFIT/GOR/ENS/93M-19

SCHEDULING AIR MOBILITY COMMAND'S CHANNEL CARGO MISSIONS

THESIS

Gregory S. Rau, Captain, USAF

AFIT/GOR/ENS/93M-19

Approved for public release; distribution unlimited

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GOR/ENS/93M-19

SCHEDULING

AIR MOBILITY COMMAND'S

CHANNEL CARGO MISSIONS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

Gregory S. Rau, B.S., M.B.A.

Captain, USAF

March 1993

Approved for public release; distribution unlimited

Thesis Approval

STUDENT: Capt Gregory S. Rau CLASS: GOR-93M

THESIS TITLE: Scheduling Air Mobility Command's Channel Cargo Missions

DEFENSE DATE: 12 March 1993

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Co-advisor	Maj John J. Borsi/ENS	John J. Borsi
Co-advisor	Lt Col James T. Moore/ENS	James T. Moore

Acknowledgements

I owe a great deal of thanks to many people. Although I cannot begin to name everyone who has helped me in some way, I would be gravely mistaken if I were to leave out my inspiration, Kim and Coleman; my mentors, LTC James Moore and MAJ John Borsi; and my constant life-saver, CPT Michael Del Rosario. Thank you all so very much!

Gregory S. Rau

Table of Contents

List of Figures

List of Tables

AFIT/GOR/ENS/93M-19

Abstract

Through the use of a linear programming model, this research revised the initial schedule for AMC's channel cargo missions to eliminate any excess delay enroute by minimizing the cumulative, weighted time-in-system for all cargo, according to a given cargo flow. In fact, the revised schedule minimizes any assigned nonnegative weighting of the time-in-system, due to the properties of equivalent measures of performance. When combined with Step One of a proposed two-step process for revising AMC's channel mission schedule, this research can be used to improve the current schedule based on Step One's cargo flow.

By carefully defining the notation and adapting the job-shop formulation, this research devised a method for modeling the scheduling of a limited-size portion of AMC's channel system and minimizing the delay enroute. If future research can improve this method using the recommendations provided, this method could become a significant part of AMC's advance planning process.

SCHEDULING AIR MOBILITY COMMAND'S
CHANNEL CARGO MISSIONS

I. Introduction

I.1 General Issue

A myriad of systems for collecting and delivering goods and services exists, ranging from transporting passengers on a bus, train, or other mode, to distributing products from factories to outlets, to collecting and disposing of refuse. The key issue relating these systems is the efficient routing and scheduling of available resources (e.g., vehicles) to meet customer demands.

Several ways for measuring schedule efficiency are available, depending on the objective of the particular problem. As Bodin observed:

> Usually the objective function is to minimize a weighted combination of capital and operating costs for the fleet [i.e., vehicles used for distribution]. It may also include a formula that represents penalties for not meeting all the time-window constraints and/or for violating other constraints. Also, vehicle routing and scheduling problems can have multiple objective criteria. Sometimes these objectives are hierarchical; in other cases, they are considered concurrently. (Bodin, 1990:574-575)

Likewise, there are several constraints which may or may not be considered in the particular problem, depending on the assumptions. For example, these constraints can include the number of vehicles, vehicle capacity, demand levels for goods or services, and restrictions on the time of delivery or collection.

The channel cargo distribution system of the United States Air Force's Air Mobility Command (AMC) is an example of a distribution system where scheduling and routing must be accomplished efficiently. And, as with any other real-world problem, the objective function and constraints can be tailored in many ways to provide the required decision-making information.

I.2 Background

One of AMC's responsibilities is managing regularly scheduled air service known as the channel network. A **channel** is a pair of airbases -- i.e., an origin and a destination, commonly called an **origin-destination (O-D) pair** -- between which AMC must fly to satisfy a military requirement. AMC provides airlift on a regular basis between O-D pairs to satisfy demand for transporting cargo; in addition, they must satisfy "frequency of visit" requirements, such as weekly visits to an embassy. Since the monthly amount of cargo requiring transport varies through the year, AMC analysts must frequently develop new schedules -- determining routes (a **route** is the path travelled by an aircraft from its departure until its return home) and number of missions (a **mission** assigns a specific type of aircraft to each route). This is no small task since there are approximately 600 channels based on cargo and 300 channels based on frequency of visit (Ackley et al, 1991:2).

AMC develops new schedules in a two phase process (see Figure 1). AMC uses a linear programming (LP) model, the Strategic Transport Optimal Routing Model (STORM), in the first phase to determine the minimum number of routes and missions needed. STORM's basic purpose is "to select the mix of routes and aircraft that will meet the monthly cargo and frequency requirements while minimizing the costs of cargo handling, military aircraft operations, and commercial aircraft leasing" (Ackley et al, undated:2). STORM provides the actual routes and missions which should be flown during the month; however, the solution to the LP model is non-integer, so AMC uses a heuristic to derive an integer set of missions. Basically, this heuristic includes all whole-number missions and any fractional missions which are cost-effective.

Analysts enter this information into a FORTRAN program, called CARGPREP, which determines a simple, monthly flight schedule by scheduling the flights of a given mission evenly throughout the month. For example, if a mission is be to flown five times in one month, CARGPREP schedules a mission every six days. The resulting, tentative schedule is used in the second phase. (The schedule is tentative because analysts at HQ AMC only use it for planning purposes and analysis; schedulers at AMC's numbered air forces develop the actual schedules manually.)

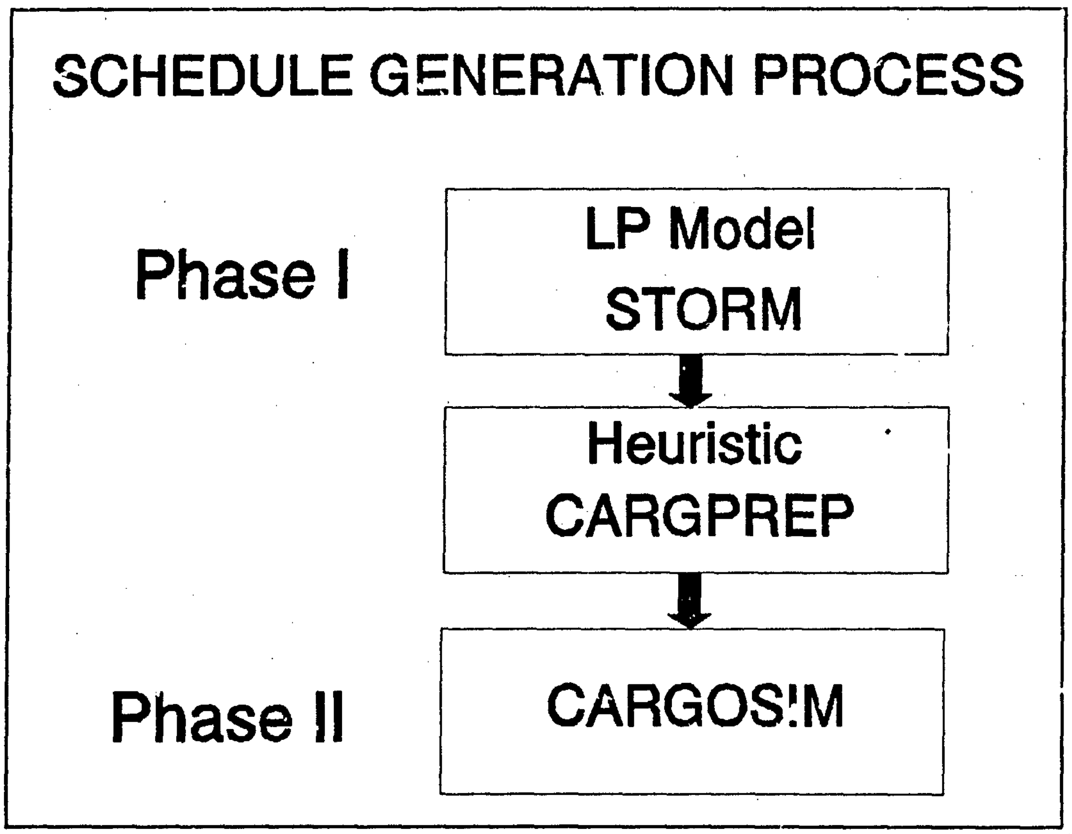

Figure 1. Current AMC Schedule Generation Process.

In the second phase, AMC uses a simulation model, CARGOSIM, to determine the amount of cargo which can be delivered "on-time" (i.e., in compliance with the Uniform Material Movement Issue Priority System (UMMIPS) standards). The CARGOSIM model uses timeliness of delivery as one of its performance measures; this measure is expressed in "average delay per cargo ton shipped between each O-D pair" (Moul, 1992: 1-5). Cargo which cannot be delivered on time is then contracted out to civilian commercial transportation. The CARGOSIM model does not determine the actual reason for insufficient cargo capacity -- i.e., whether STORM assigned too few aircraft to handle all the cargo or whether CARGPREP provided such a poor schedule that connections at transshipment points were not made.

I.3 Problem Statement

AMC currently has a process to develop tentative schedules for the channel cargo distribution system; however, the process does not guarantee a good schedule, so the AMC analysts cannot tell if transporting all cargo requires more aircraft or a better schedule. AMC needs an effective method for developing a good schedule to minimize the need for civilian carriers.

CARGOSIM measures a type of delay, but it cannot determine whether the schedule could be improved. AMC requires a method for improving the schedule as much as possible before analysts enter it into CARGOSIM. An improved schedule would allow AMC assets to ship more cargo on-time, so less would be transported by commercial means, resulting in substantial savings considering the cost of supplementing AMC airlift -- $148 million in fiscal year 1989 and $165 million in fiscal year 1988 for commercial augmentation (Ackley and others, 1991:2).

The current scheduling process is time-consuming because it takes one analyst at AMC three or four days to improve the tentative schedule using a trial and error method (Litko, 26 Aug 92). Presently, an AMC analyst uses the results of CARGOSIM to indicate large delays in the initial schedule produced by CARGPREP. The schedule is modified by changing the timetable or increasing the number of missions (Litko, 9 Sep 92). The analyst evaluates the modified schedule using CARGOSIM and re-adjusts the schedule, if necessary, continuing the process until all cargo is scheduled for delivery within UMMIPS standards. Additionally, the schedulers at the numbered air forces must go through some similar process to develop their schedules. Because of these problems associated with the current scheduling process, AMC would like a method to streamline the process.

Several methods in recent literature address various aspects of AMC's scheduling problem. These methods reduce or eliminate schedule inefficiencies such as excessive cost, insufficient use of the transporting vehicle, or ill-chosen routes. Of course, measuring schedule efficiency depends on the objective of the problem. Likewise, these methods are tailored around the objective. For example, one common objective for constructing schedules is to minimize the cost of shipping goods from the origin to the destination. Another objective is to maximize aircraft use by maximizing the number of trips assigned to each aircraft. Still another objective, and the one which this research uses, is to minimize the delay enroute.

Delay enroute is the time difference between transporting cargo directly from its origin to destination versus using other routing. There are three types of delay enroute. The first is the delay encountered when cargo is at its origin base awaiting initial transportation. The second is the delay which occurs when cargo is at an intermediate (transshipment) point awaiting transportation. The third type of delay is caused when cargo is shipped by one route when another, quicker route exists.

One proposed method to minimize the delay enroute is a two-step, iterative process (Borsi, 6 August 1992). In Step One, given any schedule, a flow of cargo is determined based on this schedule. The cargo is categorized by its quantity (weight) and its type (origin and destination). Step One determines the quantity and type of cargo that is loaded onto or taken off each aircraft as the cargo is transported from one airbase to another on its assigned path. Step Two modifies the flight departure times and revises the overall schedule based on this cargo flow. Returning to Step One with the revised schedule, the cargo flow is modified based on the new flight times. Each iteration reduces the delay enroute until the newest reduction is smaller than a predetermined value.

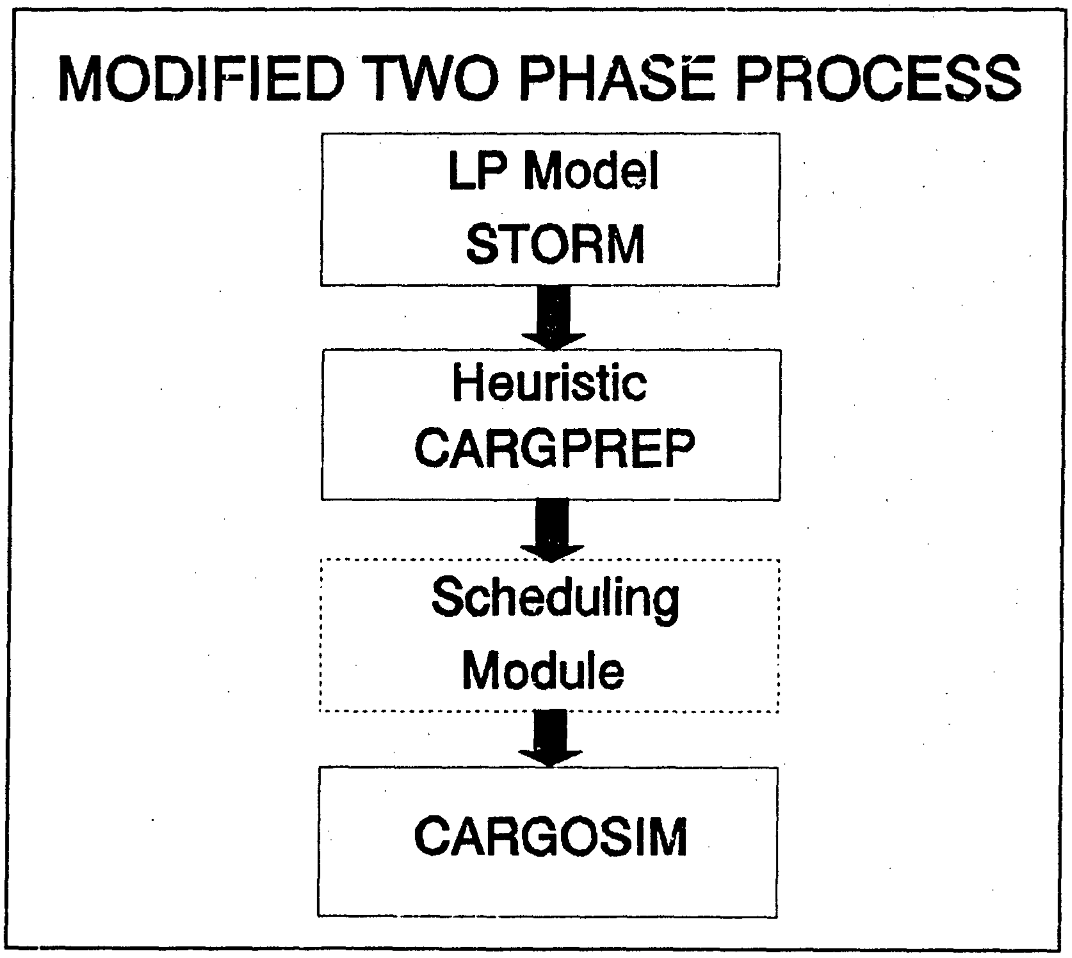

Figure 2. Modification of AMC's Two-Phase Process.

One obvious advantage of this method is that it uses the information output from STORM and the input data required by CARGOSIM. This process could be implemented after an initial schedule is created by CARGPREP (see Figure 2). This would improve that schedule before it is entered into CARGOSIM; therefore, this two-step process is compatible with the current scheduling process used by AMC. Unfortunately, no method in the current literature directly addresses this iterative approach. The problem formulations and solutions of various approaches, however, do provide insight on ways of handling diverse constraints relating to routing and scheduling problems.

I.4 Research Objective

The purpose of this research is to develop Step Two of the proposed iterative process: Given the cargo requirements and assuming a flow of cargo between O-D pairs which uses the latest schedule, modify the schedule to minimize the delay enroute. This approach is actually two-fold: the most important part is to develop a method to modify the current schedule; the other part is to determine how to measure the delay enroute for evaluating this method.

I.5 Assumptions & Scope

This research assumes that the cargo requirement for all O-D pairs is known deterministically (i.e., with certainty). AMC analysts forecast these cargo requirements based on worldwide trends.

Cargo is classified by weight only; therefore, the cargo can be divided into an infinite number of subsets. Any other characteristics such as size and urgency of need are assumed to be the same for all cargo (i.e., no outsize cargo and no priority cargo considerations). Passenger requirements will not be considered and, therefore, will not affect the amount of cargo which can be loaded.

The number of each aircraft type available is known deterministically and will remain constant (i.e., no breakdowns). Furthermore, each aircraft type has a known cargo capacity. Cargo going to different destinations may be loaded on an aircraft in any proportion provided the total weight loaded does not exceed the aircraft capacity. Any mixture of cargo is allowed on a single aircraft (i.e., no cargo is considered hazardous). Any cargo can be loaded on any aircraft (i.e., there are no restrictions for specific cargo to be loaded on specific aircraft).

Airbases are assumed to be capable of handling an unlimited amount of cargo (i.e., no restrictions on loading equipment or storage areas).

Since this research is intended to develop Step Two of AMC's proposed two-step method, a feasible cargo flow is assumed to be provided by Step One of the method. This research is not intended to develop or alter the flow of cargo through the channel system.

Maximizing the cargo load in each aircraft is of secondary importance to minimizing the delay enroute and will not be considered. Ignoring aircraft utilization is acceptable since the LP model (STORM) determines the minimum number of missions required for transporting the forecast level of cargo demand, which ensures each aircraft is cost-effective.

I.6 Definitions

The following additional terms will be used throughout this research:

leg -- the non-stop path flown between two airbases.

flight -- distinct mission and time combination; the same mission flown four times in one month will be classified as four distinct **flights**.

I.7 Format

In Chapter II, a review of literature relating to scheduling theory and the general job-shop scheduling problem will be presented and important concepts will be introduced. Chapter III covers the formulation of AMC's scheduling problem as a linear programming problem. The results of testing the formulation are discussed in Chapter IV. Chapter V presents the conclusions of this research, as well as several recommendations for future research. Finally, the computer programs developed, plus extracts of the initial data and sample output, are listed in the Appendices.

II. Literature Review

II.1 Scope and Organization of the Review

Analysts responsible for scheduling the AMC channel cargo distribution system desire a method for improving their current scheduling process. Step Two of the proposed process requires a method for finding the current optimal schedule based on the current cargo flow. Many journal articles have addressed variations of the combined "routing and scheduling" problem, but none have addressed a two-step, iterative process of this nature. Therefore, to address Step Two's goal of re-scheduling AMC's missions, this review will introduce the basics of the theory of scheduling, covering the terminology, the assumptions, and the linear programming formulation which will be referenced in later chapters.

II.2 The General Job-Shop Scheduling Problem

Scheduling is "allocating resources <u>over time</u> to perform a collection of tasks" (Baker, 1974:2). The terminology of scheduling theory "arose in the processing and manufacturing industries" (French, 1982:5). The result is a standard description of a system in which "*n* **jobs** $\{J_1, J_2, \ldots, J_n\}$ are to be processed through *m* **machines** $\{M_1, M_2, \ldots, M_m\}$" (French, 1982:5), where the "jobs" are collections of tasks and the "machines" are the resources. **Operations** are the basic tasks of which jobs consist. The time required to perform an operation is called the **processing time**. The time at which a job initially becomes available for processing is the **ready time** or **release date** of that job. Constraints which dictate the particular order of a job through the machines are called **technological constraints**. The **general job-shop problem** has "no restrictions upon the form of the technological constraints; each job has its own processing order and this may bear no relation to the processing order of any other job" [as compared to the case where all jobs have identical processing orders] (French, 1982:5).

II.3 Assumptions of the General Job-Shop

In order to introduce scheduling theory, French chooses the job-shop family because "it leads to a presentation of the theory which is particularly coherent and, furthermore, is not encumbered with a confusion of caveats and provisos needed to cover special cases" (French, 1982:15). It is his intent to explain scheduling theory in terms of the job-shop and then to allow deviations toward other contexts as the following assumptions are relaxed or dropped:

1. Each job is a single entity: no two operations of the same job may be processed at the same time.

2. No pre-emption: once a job starts on a machine, it will complete processing on that machine.

3. Each job has m distinct operations, one on each machine: no job has two operations on the same machine or skips any machine.

4. No cancellation of jobs.

5. The processing times are independent of the schedule: set-up times are sequence independent; and times to move jobs between machines are negligible.

6. In-process inventory is allowed: queues can form between machines.

7. There is only one of each type of machine: no parallel processing by any machine; and no choice of machines in the processing of a job.

8. Machines may be idle.

9. Machines never experience "down-times": no breakdowns or routine maintenance during the scheduling period.

10. The technological constraints are known in advance and are immutable.

11. There is no randomness: the quantities (e.g., the number of jobs and machines) and times (e.g., the ready and processing times) are known and fixed. (French, 1982:8-9)

Many of these assumptions apply directly to this research and were stated in Chapter I. Several others will need to be relaxed to adequately address the scheduling of AMC's channel missions in the context of the job-shop; these relaxations will be addressed in Chapter III, along with the necessary changes to the following linear programming (LP) formulation.

II.4 The Linear Programming (LP) Formulation

Using the assumptions from above and the notation which follows, the general job-shop scheduling problem with a goal of minimizing the sum of the completion times of all jobs can be formulated as the following LP problem.

Since the machine order is fixed for each job, "job j must first be processed on machine $j(1)$, then on machine $j(2)$, and so on [until processed on its last machine, $j(m)$]" (Nemhauser and Wolsey, 1988:13). Let p_{ij} denote the processing time of job j on machine i, and let t_{ij} denote the start time of job j on machine i; then $t_{j(m),j}$ denotes the start time of job j on its last machine.

Since the $(r + 1)$st operation of job j cannot start until the rth operation has been completed, Nemhauser and Wolsey present the first constraint as follows:

$$t_{j(r+1),j} \geq t_{j(r),j} + p_{j(r),j} \quad \text{for } r=1,\ldots,m-1 \text{ and all } j \qquad \textbf{(1)}$$

Since a machine can only handle one job at a time, either job j precedes job k on machine i or vice versa. By letting $x_{ijk} = 1$ if job j precedes job k on machine i, and $x_{ijk} = 0$ otherwise (where $j < k$), and by using an upper bound M on $t_{ij} - t_{ik} + p_{ij}$ for all i, j, and k, Nemhauser and Wolsey present the following disjunctive constraints:

$$t_{ij} - t_{ik} \le -p_{ij} + M(1-x_{ijk})$$
$$t_{ik} - t_{ij} \le -p_{ik} + Mx_{ijk} \qquad \forall\, i, j, and\ k \tag{2}$$

Adding to the formulation the objective function and non-negativity constraints, Nemhauser and Wolsey developed the following LP formulation of the general job-shop scheduling problem:

$$\text{MINIMIZE} \sum_{j=1}^{n} t_{j(m),j} \tag{3}$$

$$\text{SUBJECT TO} \quad t_{j(r+1),j} \ge t_{j(r),j} + p_{j(r),j} \quad for\ r=1,\ldots,m-1\ ,\forall j \tag{4}$$

$$t_{ij} - t_{ik} \le -p_{ij} + M(1-x_{ijk})$$
$$t_{ik} - t_{ij} \le -p_{ik} + Mx_{ijk} \qquad \forall\, i, j, and\ k \tag{5}$$

$$t_{ij} \ge 0 \qquad \forall i\ and\ j$$
$$x_{ijk} \in \{0,1\} \qquad \forall i, j, and\ k \tag{6}$$

$$where\ M = \max_{i,j,k} (t_{ij} - t_{ik} + p_{ij}) \tag{7}$$

This notation and formulation will be adapted and used, along with the following results, in Chapter III.

II.5 Semi-Active Timetabling and Regular Measures of Performance

Sequencing is assigning an order to a series of tasks, but a sequence "contains no (explicit) information about the times at which the various operations start and finish" (French, 1982:26). **Timetabling** is required to translate a sequence into a schedule -- it adds the time aspect for the processing of each task on each machine and for the machines' idle times or the tasks' waiting times. **Semi-active timetabling** produces a schedule in which no operation could be started earlier "without altering the processing sequence or violating the technological constraints or ready dates" (French, 1982:27). A semi-active schedule starts processing each task as soon as possible; no unnecessary idle time is inserted into the schedule.

As an example, consider three machines processing two jobs that have technological constraints which pre-determine the following sequence: job *J1*'s processing order is *M1* → *M3* → *M2*; and job *J2*'s processing order is *M2* → *M3* → *M1*. Given both jobs are ready to begin at time 0, and that machine *M1* will start by processing job *J1* and that machine *M2* will start with *J2*, the sequence can be translated to the schedule containing inserted idle time shown by the Gantt chart in Figure 3(a). The figure marks the completion times for jobs *J1* and *J2* with *C1* and *C2*, respectively; it also indicates the inserted idle times with arrows showing how much sooner any one job could begin if no other jobs were started any earlier. By removing the inserted idle time and making the schedule semi-active, both jobs finish processing as soon as possible for the given sequence [see *C1* and *C2* in Figure 3(b)].

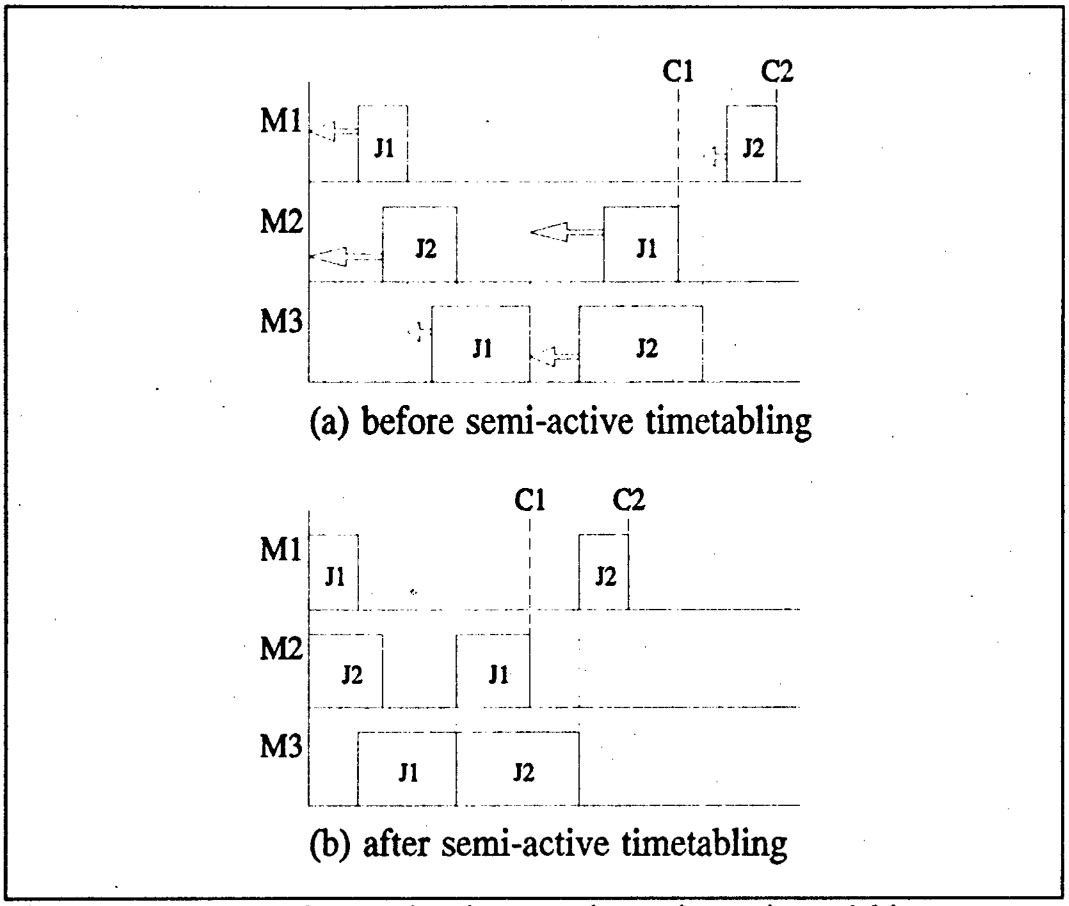

Figure 3. GANTT Chart Showing Semi-active Timetabling.

Performance measures are obviously required as the criteria for judging a schedule's success. There are a large number of complex, and often conflicting, possible objectives in scheduling: "Mellor (1966) lists 27 distinct scheduling goals" (French, 1982:9). Some typical goals are based on the average or maximum over all jobs of the **completion times** (measured from the start of the scheduling period to completion), **waiting times** (measured between operations), or **flow times** (measured from the ready date to completion). A **regular performance measure** is "one that is non-decreasing in the completion times" (French, 1982:13-4). Basically, if the completion times of one or more tasks were <u>increased</u> by a new schedule, then the regular performance measure could not be decreased.

The relationship of semi-active timetabling and a regular measure is formalized by the following:

> "**Theorem:** In order to minimize a regular measure of performance, it is only necessary to consider semi-active timetabling" (French, 1982:27).

Therefore, if a measure of performance can be proven to be regular, only semi-active schedules need to be considered to minimize this measure.

II.6 Conclusion

Combining the formulation of the general job-shop scheduling problem with the insight into semi-active schedules will be the basis of the development of this research.

III. Methodology

III.1 General

The AMC channel cargo distribution system can be viewed as a variation of the job-shop scheduling problem. As presented in Chapter II, a job-shop scheduling problem represents a system as a set of machines processing a set of jobs. The problem can have technological constraints which determine the order for any given job to be processed through the machines.

Viewing the channel cargo distribution system as a job-shop scheduling problem, a machine corresponds to an aircraft flying a single flight leg, and a job is a requirement to transport cargo from an origin to a destination. Thus, each operation is the transport of cargo across a single flight leg. The technological constraints are the ordered lists detailing the specific aircraft and flight leg combinations required to process the cargo, as determined by the cargo flow algorithm in Step One.

This chapter first examines the use of delay enroute as the measure of performance. Then, through a small example problem, concepts and notation to be used in the linear programming (LP) formulation will be developed. Since size is a significant concern for any large-scale problem such as scheduling AMC's missions, it will be addressed both in general terms and specifically for the channel system.

III.2 Performance Measure

The goal of the proposed two-step scheduling process is to minimize the delay enroute. The total delay enroute is calculated as the difference between the total time a piece of cargo spends in the system and the minimum time which would be required to transport the cargo across the quickest path. Since this minimum time is a fixed value which could be determined for any Origin-Destination (or O-D) pair, minimizing the total time-in-system will also minimize the delay enroute. These are **equivalent** measures, as defined by French: "two performance measures are equivalent if a schedule which is optimal with respect to one is also optimal with respect to the other and vice versa" (French, 1982:28). The benefit of using time-in-system is that it can be calculated directly by using a piece of cargo's completion time and time of creation without concern for calculating the duration of the quickest path for that cargo's O-D pair.

The time-in-system should be weighted by the size of the cargo to place greater significance on the larger shipments. This is directly in line with the current performance measure of "average delay per cargo ton shipped between each O-D pair" (Moul, 1992:1-5) used by AMC's CARGOSIM model as a measure of timeliness.

Cumulative weighted time-in-system is a **regular measure of performance**, as defined in Chapter II. To prove this, consider cargo piece j. Let w_j be the size, C_j be the completion time, and r_j be the ready time of cargo piece j. Since the time-in-system of piece j is $(C_j - r_j)$, the cumulative weighted time-in-system is denoted

$$\sum_{j=1}^{n} w_j(C_j - r_j) = w_1(C_1 - r_1) + w_2(C_2 - r_2) + \dots + w_n(C_n - r_n) \qquad \textbf{(8)}$$

Compare two schedules, S and S', of the same n jobs, where schedule S has completion times $C_1, C_2, \dots, C_n$, schedule S' has completion times $C_1', C_2', \dots, C_n'$, and the following is true:

$$C_1 \le C_1', \ C_2 \le C_2', \ \dots, \ C_n \le C_n' \qquad \textbf{(9)}$$

Then, since $w_j > 0$ for all j, and since both w_j and r_j are known and fixed for all j, the following statements must be true:

$$C_1 - r_1 \le C_1' - r_1, \ \dots, \ C_n - r_n \le C_n' - r_n \qquad \textbf{(10)}$$

$$w_1(C_1 - r_1) \le w_1(C_1' - r_1), \ \dots, \ w_n(C_n - r_n) \le w_n(C_n' - r_n) \qquad \textbf{(11)}$$

This last statement implies that cumulative weighted time-in-system is indeed non-decreasing in completion times:

$$w_1(C_1 - r_1) + w_2(C_2 - r_2) + \ldots + w_n(C_n - r_n) \leq w_1(C_1' - r_1) + w_2(C_2' - r_2) + \ldots + w_n(C_n' - r_n) \tag{12}$$

Therefore, based on the theorem stated in Chapter II, only semi-active timetabling needs to be considered (i.e., all operations should start as soon as they can) when the goal is to minimize cumulative weighted time-in-system. To better understand this concept, consider the following example.

III.3 Example Problem

Setting up and solving a miniature version of AMC's scheduling problem sheds light on the concepts, notation, and formulation proposed. Consider the system in Figure 4, having three airbases (A, B, and C) and two missions (where "mission one" is flying from A to B and then returning to A, and "mission two" is flying from C to B and back to C). In this system, all cargo being transported from A to C or vice versa must transship through base B; one transshipment point provides a sufficient example since AMC's STORM limits cargo to a single transshipment.

Assume only one aircraft is available for each of the two missions, where both aircraft are the same type and have sufficient capacity for all assigned cargo (aircraft capacity is a concern for flowing the cargo through the system but not for re-scheduling the flight legs). Each

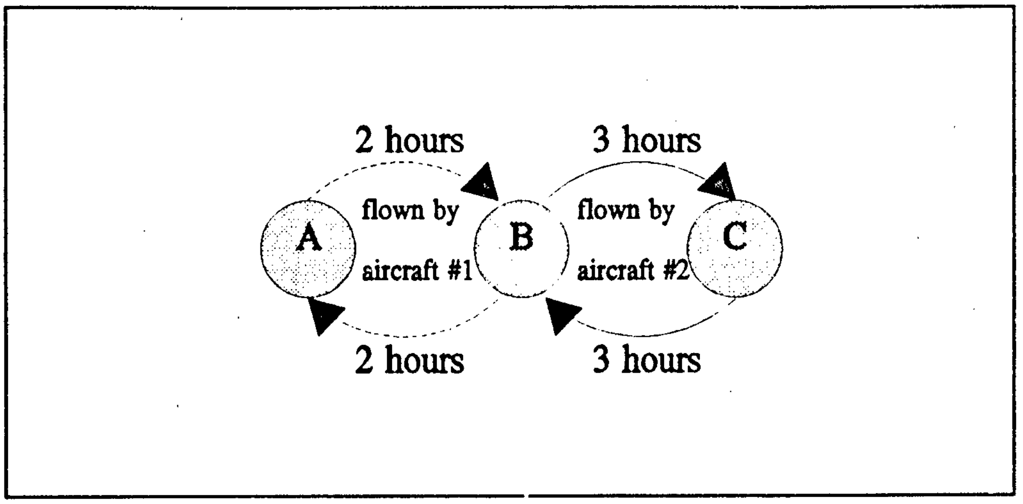

Figure 4. Three-airbase Example.

distinct leg flown is given a unique label -- for example, the leg flown from A to B is labeled "flight leg L1" the first time it is flown and "flight leg L5" the second time. Assume the standard ground time is one hour (for refueling, etc.) and the flight times are as follows: two hours from A to B or from B to A; and three hours from C to B or from B to C. If both missions are flown twice (i.e., a total of four flights) in a 20-hour time period, the second flight of a mission is initially scheduled for 10 hours after the first flight, mimicking AMC's CARGPREP; also similar to CARGPREP's method, the first flight of each mission was scheduled arbitrarily -- see the departure and arrival times for all eight flight legs (each flight has two distinct flight legs) in Table 1.

Flight Leg Number	Origin & Destination	Departure Time	Flight Time	Arrival Time
L1	A - B	2:00	2 hours	4:00
L2	B - A	5:00	2 hours	7:00
L3	C - B	1:00	3 hours	4:00
L4	B - C	5:00	3 hours	8:00
L5	A - B	12:00	2 hours	14:00
L6	B - A	15:00	2 hours	17:00
L7	C - B	11:00	3 hours	14:00
L8	B - C	15:00	3 hours	18:00

Table 1. Initial Flight Schedule for Example System.

For this example, assume sixteen pieces of cargo need to flow through the system in a timely manner, so the goal is to minimize the cumulative time-in-system. (The weight of each piece can be ignored if we assume all of these pieces are the same size -- for example, one-ton pieces.) To further simplify this example, assume instantaneous loading and unloading of all cargo at origin, destination, or transshipment points; while this may not seem realistic, these times would normally be aggregated, along with the flight times and ground times of the aircraft, into the processing times for the cargo.

After assigning arbitrary ready times for the sixteen pieces of cargo listed in Table 2, the cargo is flowed through the system as quickly as possible (for the current schedule) by loading each piece on the next available aircraft flying to the piece's destination or the transshipment point (base B). In this system, the pieces that are transshipped need to be transported across two flight legs, while the rest can be transported directly from origin to destination across a single flight leg. By tracking the time when cargo reaches its destination, the Time-in-System (in hours) can be computed as the Finish Time of the cargo's last leg minus the cargo's Ready Time, as shown in Table 2.

While setting up this example and flowing the cargo, several points became clear. First, note that more than one piece of cargo may be transported by a given flight leg. Since a machine represents an aircraft flying a flight leg, this is equivalent to having a machine process multiple tasks (of different jobs) simultaneously. While this seems obvious enough, it conflicts with the seventh assumption of the job-shop (see Chapter II), which assumes no parallel processing by any machine. This conflict could be addressed using **multiprocessor scheduling** (French, 1982:200), which would allow the tasks to "choose" between identical

Cargo Piece #	Cargo O-D Pair	Cargo Ready Time	Cargo Flow 1st Leg	Cargo Flow 2nd Leg	Finish Time 1st Leg	Finish Time 2nd Leg	Time in Sys.
1	A - C	0:00	L1	L4	4:00	8:00	8
2	A - C	1:00	L1	L4	4:00	8:00	7
3	A - C	5:00	L5	L8	14:00	18:00	13
4	A - C	6:00	L5	L8	14:00	18:00	12
5	A - C	9:00	L5	L8	14:00	18:00	9
6	C - A	0:00	L3	L2	4:00	7:00	7
7	C - A	3:00	L7	L6	14:00	17:00	14
8	C - A	4:00	L7	L6	14:00	17:00	13
9	C - A	6:00	L7	L6	14:00	17:00	11
10	C - A	7:00	L7	L6	14:00	17:00	10
11	A - B	0:00	L1	-	4:00	-	4
12	B - A	3:00	L2	-	7:00	-	4
13	A - B	7:00	L5	-	14:00	-	7
14	C - B	0:00	L3	-	4:00	-	4
15	B - C	4:00	L4	-	8:00	-	4
16	C - B	6:00	L7	-	14:00	-	8

Table 2. Initial Cargo Flow for Example System.

machines; however, these "identical machines" are actually a single aircraft (with a single departure time for the given flight leg) which <u>must</u> process all assigned tasks simultaneously. Therefore, this conflict can be handled by considering all cargo pieces that are assigned to a single flight leg to be a single task, using only the most restrictive ready time as the limit for how early the aircraft can depart.

A piece of cargo's "ready time" is the time at which it arrives at an airbase and is ready to be transported across its first leg. The "finish time" of its first leg must be used to determine when this cargo is ready to be transported across its second leg. By combining French's definition of **ready times** (which refers to when jobs are <u>initially</u> <u>ready</u> to begin processing) with Nemhauser and Wolsey's notation for **start times** (which refers to when jobs <u>actually</u> <u>start</u> being processed), the time when cargo j is ready to be transported across flight leg i will be defined as the **available start time** of job j on machine i and denoted $r_{i,j}$.

The next observations are that no job requires all machines and that the jobs may actually require different numbers of machines -- for example, piece #1 requires two machines (L1 and L4), but piece #16 requires only one (L7). Instead of each job being processed on each of m machines, job j will be processed on m_j machines, where m_j will be between 1 and m depending on the requirements of job j. While this violates the third assumption of the general job-shop, it only complicates the notation and the bookkeeping, and not the complexity of the problem -- as French admits, "In short, we made this assumption purely for tidiness" (French, 1982:199).

Finally, the current schedule has inserted idle time -- operations could start earlier without violating ready times and technological constraints; therefore, the schedule is not semi-active. Since the goal is to minimize a regular performance measure, the theorem in Chapter II ensures only semi-active schedules need to be considered. Tables 3 and 4 show the results of making the current schedule semi-active.

Transforming the initial schedule into a semi-active schedule reduces the cumulative time-in-system for the sixteen pieces of cargo from 135 hours down to 101 hours. This sizeable improvement in timeliness resulted from starting each operation as soon as possible. The earliest starting time for each operation was based on two primary restrictions: the aircraft must be available to process the cargo; and all assigned cargo must be available for that operation. Once both of these requirements were met, the aircraft departure was scheduled to eliminate any extra

Flight Leg Number	Origin & Destination	Departure Time	Flight Time	Arrival Time
L1	A - B	1:00	2 hours	3:00
L2	B - A	4:00	2 hours	6:00
L3	C - B	0:00	3 hours	3:00
L4	B - C	4:00	3 hours	7:00
L5	A - B	9:00	2 hours	11:00
L6	B - A	12:00	2 hours	14:00
L7	C - B	8:00	3 hours	11:00
L8	B - C	12:00	3 hours	15:00

Table 3. Semi-active Flight Schedule for Example System.

Cargo Piece #	Cargo O-D Pair	Cargo Ready Time	Cargo Flow		Finish Time		Time in Sys.
			1st Leg	2nd Leg	1st Leg	2nd Leg	
1	A - C	0:00	L1	L4	3:00	7:00	7
2	A - C	1:00	L1	L4	3:00	7:00	6
3	A - C	5:00	L5	L8	11:00	15:00	10
4	A - C	6:00	L5	L8	11:00	15:00	9
5	A - C	9:00	L5	L8	11:00	15:00	6
6	C - A	0:00	L3	L2	3:00	6:00	6
7	C - A	3:00	L7	L6	11:00	14:00	11
8	C - A	4:00	L7	L6	11:00	14:00	10
9	C - A	6:00	L7	L6	11:00	14:00	8
10	C - A	7:00	L7	L6	11:00	14:00	7
11	A - B	0:00	L1	-	3:00	-	3
12	B - A	3:00	L2	-	6:00	-	3
13	A - B	7:00	L5	-	11:00	-	4
14	C - B	0:00	L3	-	3:00	-	3
15	B - C	4:00	L4	-	7:00	-	3
16	C - B	6:00	L7	-	11:00	-	5

Table 4. Cargo Flow with Revised Times for Example System.

delay. For a small example system such as this, the process of revising the flight schedule without violating requirements of the cargo and the aircraft can be done manually; however, for a scheduling problem as large as AMC's channel system, the interaction of the requirements becomes far too complicated to make the necessary revisions by hand. In the next section, this problem of revising the schedule can be formalized as an LP model.

III.4 Linear Programming Formulation

By using notation from the previous section, borrowing some of Nemhauser and Wolsey's notation from Chapter II, and defining some new terms, AMC's scheduling problem can be formulated as an LP problem.

Recall that j is the index of a single "piece" of cargo (where that piece may be anything from one box to a large number of boxes, crates, and miscellaneous items) with a known Origin-Destination designation (referred to as the cargo's **O-D pair**). The first flight leg used to transport cargo j is designated $j(1)$; the second leg, $j(2)$; and so on, through the cargo's last leg, $j(m_j)$, where m_j is the total number of flight legs used to transport cargo j. As an example, if piece j is to be transported across flight legs $L3$, $L4$, $L5$, and $L9$ (so, $m_j = 4$), then $j(1) = L3$, $j(2) = L4$, $j(3) = L5$, and $j(4) = L9$.

Define a set, D_i, for each flight leg *i*, to include the index of each piece of cargo to be transported across that leg. For example, if cargo pieces *7*, *12*, and *23* all require transport across flight leg *L8*, then $D_{L8} = \{7, 12, 23\}$.

The **size** (in tons) of cargo *j* is denoted by w_j and is a known parameter for all pieces.

As defined in Section III.3, $r_{i,j}$ is the **available start time** of cargo *j* on flight leg *i*. Any cargo piece's initial available start time, $r_{j(1),j}$, is assumed to be known and fixed, based on AMC's monthly forecast. Successive available start times, $r_{j(2),j}$ through $r_{j(mj),j}$, will depend on the departure times of the previous flight legs and the processing times of the cargo, as presented below. The **completion time** for cargo *j* is the time cargo *j* would be available to start an additional leg, denoted $r_{j(mj+1),j}$; this additional leg is beyond leg $j(m_j)$ and designated $j(m_j+1)$ = *"end"* for all *j*.

The **processing time of cargo** *j* on flight leg *i*, $p_{i,j}$, includes all required processing -- i.e., the flight time of leg *i* plus the ground times for refueling, loading cargo, unloading cargo, or any required combination. The processing time may differ for two cargo pieces being transported across the same flight leg if, for example, one piece remains on the aircraft for its next leg while the other is unloaded from this aircraft and loaded onto a different aircraft. A similar parameter is the **processing time of the aircraft** flying flight leg *i*, denoted by P_i. While this parameter will have the same value as $p_{i,j}$ for any cargo which is transported by consecutive legs *i* and *i+1*, it is independent of the number of pieces loaded or unloaded at a stop -- ground times are based on AMC's determination of an aircraft's requirements, and the loading and unloading of cargo is done concurrently with the refueling, mission planning, and other ground activities. Because this entire system is assumed to be deterministic, all processing times are assumed to be known in advance and fixed, based on the assigned cargo flow or the requirements of the aircraft and crew.

Defining TO_i as the **Take-Off time** of flight leg *i* (where every leg of every flight has a unique designation) provides a convenient and necessary way of distinguishing between the available start times of the individual pieces assigned to leg *i* and the actual start time of these tasks. (The actual start time for all of these tasks is, of course, the Take-Off time, TO_i.)

Define a set, *F*, to contain all flight legs which have a preceding leg in the same flight. If flight legs *i-1* and *i* are consecutive legs of the same *flight* (i.e., leg *i-1* and leg *i* are flown by the same aircraft during a single mission), then flight leg *i* is in the set *F*; therefore, set

F contains all legs except the *first* leg of every flight. This is a way of tracking whether or not two consecutively numbered legs were flown by the same aircraft, since the flight legs are numbered sequentially as shown in the example in Section III.3.

The LP model of AMC's scheduling problem can now be written as follows:

OBJECTIVE FUNCTION:

$$MIN \sum_{j} [w_j \times (r_{j(m_j+1),j} - r_{j(1),j})] \qquad \textbf{(13)}$$

SUBJECT TO

$$r_{j(s+1),j} = TO_{j(s)} + p_{j(s),j} \quad \forall j,\ and\ s=1,\ldots,m_j \qquad \textbf{(14)}$$

$$TO_i \geq \max_{j \in D_i} r_{i,j}\ , \qquad \forall i \qquad \textbf{(15)}$$

$$TO_i \geq TO_{i-1} + P_{i-1}\ , \qquad \forall\ i \in F \qquad \textbf{(16)}$$

$$TO_i \geq 0\ , \qquad \forall\ i \notin F \qquad \textbf{(17)}$$

where $j(s) = i$ if the sth leg for cargo j is flight leg i

[Note: nonnegativity of the **available start times** follows automatically from the definitional constraints in Eq (14) along with the nonnegative parameters, $p_{j(s),j}$ and $r_{j(1),j}$, and nonnegative variables, $TO_{j(s)}$, for all j]

The goal for this problem is to minimize the Time-in-System for all cargo, weighted by the size of the cargo.

The decision variables are the **Take-Off times** of each flight leg and the **available start times** for each leg (after the initial leg) for each piece of cargo. The **processing times, sizes**, and <u>initial</u> **available start times** are known parameters.

The **available start time** of each piece of cargo across each leg is actually the time the piece arrives at a base (after all required processing). This arrival time is determined by the previous leg's Take-Off time plus the processing time, which incorporates the flight time across the leg and either the ground time of the aircraft (if the cargo remains onboard that aircraft) or the unloading and handling time of the cargo (if the cargo is being transshipped). This means a piece is not ready to be transported across a leg until it has departed its preceding base and been fully processed. Since this defines the **available start time** of each piece of cargo for each leg, Equation (14) is classified as a definitional constraint.

Equation (15) prevents an aircraft from departing before all of its cargo has arrived: the **Take-Off time** cannot be earlier than the **available start time** of the latest piece of cargo. For any given flight leg i, Equation (15) actually represents a series of equations, with one equation for each piece of cargo in set D_i. Only one of these constraints can be binding, as the earlier **available**

start times do not limit the **Take-Off time** as much as the latest one; however, these **available start times** are generally unknown before solving the problem, unless leg i happens to be the first leg for all of the pieces in set D_i [$r_{j(1),j}$ are known].

The **Take-Off time** of leg i is further restricted by the **Take-Off** and **processing times** of the previous leg, $i-1$, <u>if</u> both legs are flown by the same aircraft. (Only if the same aircraft flies these consecutively numbered legs will i be an element of the set F.) Therefore, the constraint labeled Equation (16) prevents an aircraft from departing on a given leg until after it has departed its previous base, flown the previous flight leg, and been serviced for this flight leg.

The nonnegativity constraints in Equation (17) ensure the first leg of any flight will not start before the beginning of this scheduling period.

III.5 Problem Size of the LP Formulation

The total size of the scheduling problem when using the LP formulation is a function of the total number of pieces of cargo, the number of flight legs used to transport each piece, the total number of flight legs to be flown, and the number of cargo pieces on each flight leg. In order to express the problem mathematically, define K to be the total number of pieces of cargo (so, j = 1, 2, ..., K), and define

N to be the total number of flight legs in this scheduling period (so, $i = 1, 2, \ldots, N$).

The number of decision variables representing the **Take-Off times** is simply the number of distinct flight legs, N. The number of **available start time** decision variables is the sum of the m_j over all j, since **available start times** must be determined for $j(2)$ through $j(m_j+1)$ for each j (recall that the **available start time** of $j(m_j+1)$ is equivalent to that cargo's completion time). Therefore, the total number of decision variables in the LP formulation is

$$N + \sum_{j=1}^{K} m_j \qquad \textbf{(18)}$$

The number of definitional constraints specified by Equation (14) is the same as the number of **available start time** decision variables: sum of the m_j over all j. Equation (15) specifies one constraint for each element of D_i for all i; so, the total number of these constraints is the sum of the cardinality of D_i over all i. The total number of constraints required by Equations (16) and (17) is N, since the **Take-Off time** of each flight leg is <u>either</u> in the set F <u>or</u> not [i.e., in Eq (16) <u>or</u> in Eq (17)]. Therefore, the total number of constraints in the LP formulation is

$$\sum_{j=1}^{K} m_j + \sum_{i=1}^{N} |D_i| + N \tag{19}$$

This function, however, can be simplified by noting that every element of D_i must have a corresponding increment to the number of flight legs to be used by a piece of cargo. For example, adding flight leg *L4* to those required by cargo piece *j* not only adds piece *j* to set D_{L4}, but also increases the value of m_j by one. Therefore, summing the cardinality of D_i for all *i* yields the exact same results as summing the m_j over all *j*, which means the total number of constraints in the LP formulation can now be written as follows:

$$N + 2*(\sum_{j=1}^{K} m_j) \tag{20}$$

To be meaningful, of course, these total numbers of the variables and constraints in the LP formulation must be put into the context of a real problem. In particular, the size of AMC's scheduling problem must be determined.

III.6 Problem Size of AMC's Channel Cargo System

The total size of the LP formulation required to model the scheduling of the entire AMC channel cargo system is an important consideration due to current computer limitations. Computers available to AMC's Force Structure Analysis office are capable of solving an LP problem with as many as 160,000 variables and 20,000 rows [i.e., constraints] (Whisman, 30 October 1992). Since the LP formulation developed above has more constraints than variables, the limit of 20,000 rows will be the upper bound for the size of the problem that AMC computers can handle.

Based on the sample data provided by AMC, including the output from STORM and CARGPREP which provide the routes and number of missions in 'ROUTE.DAT'(Appendix D) and 'SCHEDULE.RAW'(Appendix E), AMC aircraft and contracted civilian aircraft fly a total of 607 flights, amounting to 2757 distinct flight legs, for one month of the channel cargo system. Therefore, $N = 2757$.

The number of pallet positions for each aircraft type can be used as the upper bound for the number of distinct pieces of cargo transported across any given leg (Litko, 1 December 1992). Multiplying this number by the aircraft utilization by plane type from CARGOSIM's output, 'JET.DAT', provides a better estimate of the number of pallet positions actually used (for now, each pallet will be assumed to be a unique piece of cargo). Taking this average number of pallet positions occupied per leg for a given aircraft type and multiplying by the number of flight legs scheduled for that aircraft yields an estimate of the product [(number of pieces per leg) x (number of legs)] -- i.e., the estimated number of **piece-legs**; this product, when summed over all of the aircraft types, is an estimate of the sum of the D_i over all i (also an estimate of the sum of the m_j over all j). For one month of the entire channel cargo system, this estimate was computed to be 26,632 (see Appendix T).

Since the total number of constraints is N plus twice the sum of the m_j over all j, the LP formulation of one month of the channel system would require approximately 56,000 constraints. This number far exceeds the current computer capability of AMC, so the LP formulation for scheduling the entire channel cargo system over a full month cannot be solved at present.

III.7 Reduction of the Problem Size

Since the LP formulation of the full system for a full month is too large, consider breaking the system into separate theaters and limiting the planning horizon to reduce both the number of flight legs and the number of cargo pieces in the problem. If the resulting smaller problems are sufficiently independent, each of these could be solved and the solutions combined.

Dividing the channel cargo system into separate theaters (distinct geographic areas) seems feasible, especially since this was the method formerly used by STORM (Whisman, 1 December 1992). Since the channel system is based on transporting cargo from the United States to other parts of the world, the system should be divided according to the amount of interaction between the airbases in the United States and in other parts of the world. This interaction is in the form of connecting routes and shared O-D pairs. A natural way to divide the channel cargo system then is to have four distinct theaters which include their interaction with the U.S.: the Pacific, including Australia, New Zealand, Japan, Korea, and Indonesia; Europe and Southwest Asia, including Iceland and Greenland; Africa, including Diego Garcia; and the Americas, including Canada, the Caribbean, Central America, and South America (Litko, 13 Oct 1992).

The four theaters have substantially more interactions with the U.S. than with each other. This was documented in a recent AMC study containing 435 O-D pairs, which consisted of 176 pairs within the Pacific theater, 147 pairs in the Europe and Southwest Asia theater, 92 pairs within the Americas, and 11 in the African theater; the only interactions between theaters were a single O-D pair between Europe and Africa, and eight pairs between Europe and the Pacific region (Whisman, 27 Oct 1992). Considering the sizes of the theaters, this interaction seems insignificant.

The remainder of this research will concentrate on a single theater to reduce the problem size. The Europe and Southwest Asia theater (to be referred to as simply the European theater from this point forward) was chosen due to its characteristics of size and interactions. The European and Pacific theaters are considerably larger than the other two in terms of the amount of cargo, the number of routes, and the length (referring to the number of stops) of these routes (Robinson, 22 September 1992). The interactions within the European theater cause a higher chance for transshipment of the cargo than in the Pacific theater (Whisman, 22 September 1992); these transshipments are important because they cause the interactions between the different flights.

Analyzing the size of the LP formulation for one month of the European theater in the same way as done above for the entire system, the total number of flight legs (*N*) is 1228, and the estimate of the number of *piece-legs* becomes 12,234 (see Appendix U). Doubling this last number and adding *N* provides an estimate of approximately 25,700 for the number of constraints needed in the LP problem. Since this value still exceeds AMC's computer capabilities, the next step is to consider reducing the planning horizon.

Although normal AMC studies cover a planning horizon of 30 days (Whisman, 22 September 1992), AMC analysts forecast the cargo generation for one week and then assume the cargo is generated in the same manner each week through the month. By assuming the channel missions were developed to handle this pattern of cargo generation, the time window for this problem can also be reduced to one week. A seven-day planning horizon reduces the number of constraints in the problem to approximately one-fourth of the original number, or an estimate of 6,425 for the European theater. This is well within the computer capabilities of AMC. Note that based on using a time horizon of one week, the model of the entire channel system would be small enough to be solved on AMC's computers. To coincide with research being done on the cargo flow algorithm for Step One of the proposed method (Del Rosario, 1993), and to work within the existing computer capabilities at AFIT, this research will only consider the scheduling of missions for one week in the European theater.

III.8 Modeling the European Theater as an LP Problem

One week of the channel missions in the European theater involves 40 airbases (Appendix A) shipping cargo designated by 140 O-D pairs within and between Europe, Southwest Asia, and the United States across 49 different routes -- for a total of 81 flights comprised of 377 distinct flight legs (Robinson, 22 September 1992). Although AMC analysts have forecast the cargo generation for a one-week period (Appendix B), these forecasts need to be converted to distinct pieces of cargo for this formulation (see 'DEMAND.FOR' in Appendix C). After conversion into pieces ranging in size from less than one ton to a limit of five tons (which is the standard weight limit for a cargo pallet), the cargo in this system is represented by 883 distinct pieces (Appendix C).

A very basic assumption of this research is that the flow of the cargo through the system is determined prior to the attempt to adjust the schedule in Step Two. Since an actual cargo flow for this system is not yet available, and since this research requires only a feasible cargo flow, the 883 distinct cargo pieces were processed through a FORTRAN program which combined AMC's current schedule, available flights, and transshipment points for one week in this theater ('CARGFLOW.FOR' in Appendix Q). Although this procedure was unable to flow all of the cargo through the system, it did flow 609, 666, or 621 pieces, depending if the list of cargo was read forward, backward, or sorted by time of creation, respectively (Appendix K). Since any feasible flow can be used as the starting point of the LP formulation, the unflowed cargo was deleted, and the three sets of successfully flowed cargo became the focus for further development. In reality, the unflowed cargo would have to be transported by additional AMC missions or contracted out to commercial carriers.

These pieces of cargo, along with their respective lists which detail the order of flight legs needed to transport the cargo from origin to destination (Appendix L), became the input of another FORTRAN program, 'SCHEDMPS.FOR' (Appendix R). This program creates a Mathematical Programming System (MPS) format of the LP formulation; the MPS format (Schrage, 1987:41-44) is a standard format for transferring LP problems into a commercial solver such as MINOS (Modular In-core Nonlinear Optimization System). MINOS was then successfully used to solve the scheduling problem for one week of the European theater, using each of the three feasible cargo flows discussed above. Chapter IV presents the results of the testing.

IV. Results

IV.1 General

The actual success of modeling the AMC channel cargo distribution system with the LP formulation must still be established. If the formulation performs as expected, and can be proved to hold in all circumstances, Step Two of AMC's proposed two-phase method for improving their advance planning schedule is complete and ready to be joined with Step One. Still, key benefits and weaknesses of this research must be discussed to enable a successful marriage of the two steps.

IV.2 Results of the Sample Cargo Flows

The three cargo flows (each for the same week of the European theater) introduced at the end of Chapter III were individually formulated and then solved. The solutions from MINOS provide the **Take-Off times** for each of the 377 flight legs during this one week, the **available start times** for every leg that each piece of cargo travels, and the value of the objective function -- minimizing the Weighted Time-in-System, or WTIS (see Appendix M). The most basic comparison to be made (between the initial schedule and the one improved by the LP) is a check of the cumulative WTIS and the effects on the individual pieces of cargo. Table 5 shows this comparison and confirms that the LP did, in fact, make a noticeable improvement in the timeliness of the cargo delivery.

Description	Flow #1	Flow #2	Flow #3
Cum. Weighted TIS before LP solution	118213.39 ton*hrs	119534.92 ton*hrs	131121.27 ton*hrs
Cum. Weighted TIS after LP solution	90425.55 ton*hrs	97531.76 ton*hrs	97101.16 ton*hrs
REDUCTION in Cum. Weighted TIS	27,787.84 ton*hrs	22,003.16 ton*hrs	34,020.11 ton*hrs
Total Number of Cargo Pieces	609 pieces	666 pieces	621 pieces
Avg. Reduction in WTIS per piece	45.63 ton*hrs	33.04 ton*hrs	54.78 ton*hrs
Avg. Weight of each Piece	3.02 tons	3.00 tons	2.99 tons
Avg. Reduction in TIS per Piece	15.11 hours	11.01 hours	18.32 hours
Original value for TIS per Piece	64.29 hours	59.87 hours	70.60 hours
Percent Improvement in Avg TIS (by LP)	23.50%	18.39%	25.95%

Table 5. Effect of LP on Time-in-System for 3 Cargo Flows.

Since Cargo Flow #1 provides results which fall between the other two, it serves as a good specimen for further study. Categorizing the LP's improvement in individual cargo's Time-in-System (or TIS) will help to see if this procedure can make enough difference to warrant its use.

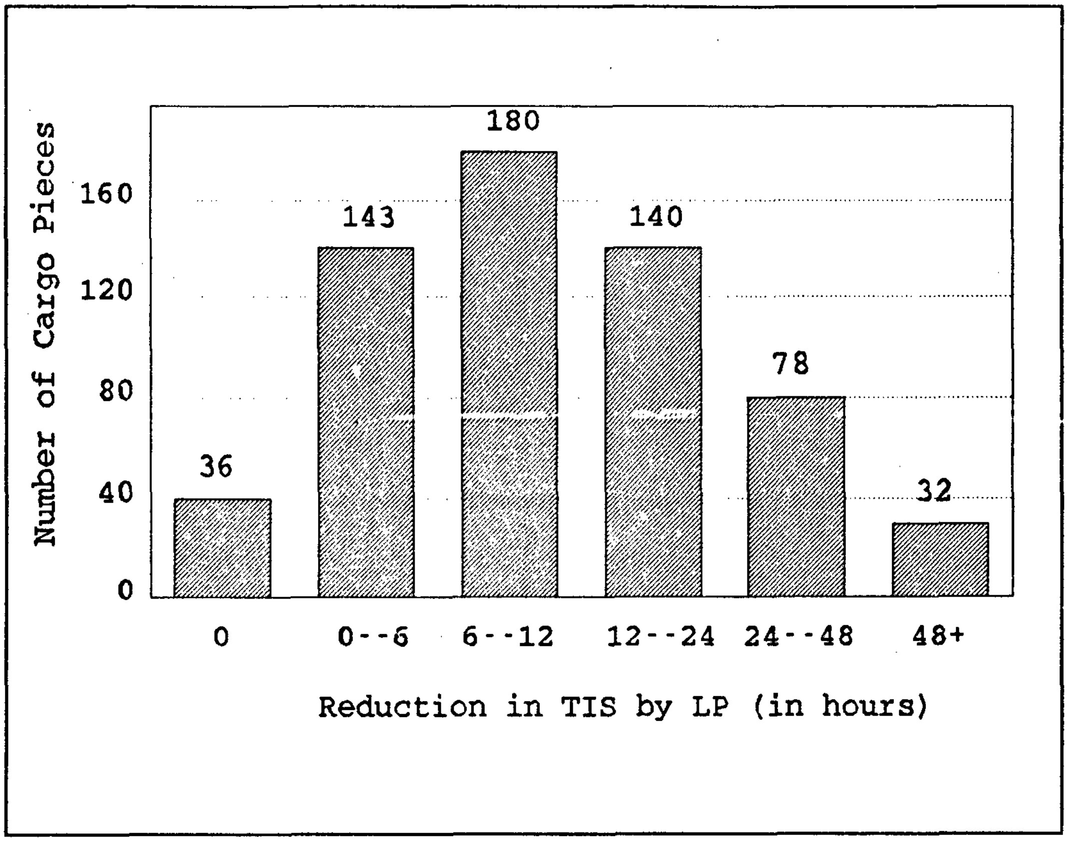

Figure 5. Histogram of TIS Reduction for Data Set #1.

The histogram in Figure 5 shows that the TIS for 359 of the 609 pieces was reduced by less than 12 hours; however, this means the remaining 250 pieces were scheduled for delivery at least 12 hours earlier than in the original schedule. Of these, 32 pieces (5% of the cargo pieces) had reductions in TIS of at least 48 hours; closer examination reveals all of these TIS reductions were at least 60 hours (see Appendix S). The maximum improvement in TIS for any one piece in

Cargo Flow #1 was 86.4 hours for pieces #257 and #262 -- this reduction is quite significant considering their original TIS was 170 hours. Overall, the LP solution provides a good deal of improvement in the timeliness of the cargo delivery for this, and the other two, initial cargo flows.

While these results are impressive, considering the amount of improvement in the objective functions alone is not sufficient. The next section will examine the LP formulation in more depth.

IV.3 Proving the Validity of the LP Formulation

Proof that this method will perform as expected under all circumstances is necessary. This research (Step Two) is expected to adjust a schedule to minimize the delay enroute and, therefore, the time-in-system for any given cargo flow. Proof that the method will work hinges on the basic assumption that the cargo flow is, indeed, feasible.

Since each *machine* (i.e., an aircraft flying a single flight leg) processes a single, aggregated *operation* (i.e., transporting all assigned cargo across that leg), only one possible processing sequence can exist for the given cargo flow. This cargo flow details which cargo pieces are to be transported across a given flight leg and, more importantly, the exact order of flight legs to be used by any one piece of cargo. These ordered lists, along with the necessary, sequential relationship of flight legs for any given aircraft, become the technical constraints limiting the possible start time for a flight leg.

For any given sequence, semi-active timetabling produces a unique schedule (French, 1982:28). Since only a single sequence exists for any given cargo flow, and since weighted time-in-system is a regular performance measure (see Section III.2), semi-active timetabling produces the only schedule to be considered while incorporating all of the technical constraints. The process of arriving at this semi-active schedule is the entire purpose of the LP formulation.

The only question that remains, then, is whether the LP formulation and, in particular, the constraints of the LP adequately address the required technical constraints. Since an aircraft cannot *process* (i.e., transport) the single, aggregated cargo until the last piece of this cargo arrives (i.e., is available), the **Take-Off time** cannot be earlier than the latest **available start time** for any of this flight leg's cargo. Equation (15) in Chapter III guarantees this restriction through use of the definitional constraints for **available start times** in Equation (14).

The other major consideration for determining how early an aircraft can depart on a flight leg is based on the completion of the previous flight leg (unless this is the first leg of a flight). Equation (16) guarantees that the **Take-Off time** for a given flight leg cannot be prior to the completion of the previous flight leg, and Equation (17) prevents the first leg of a flight from departing before the beginning of this planning horizon. Provided all of the parameters of this model are properly defined and entered into the problem, this LP formulation will guarantee the optimal schedule for minimizing the time-in-system (and, therefore, the delay enroute) for any given, feasible cargo flow.

Extensive study of the LP solutions for the three sample cargo flows was performed to confirm the accuracy of the formulation. Comparing a large sample of Take-Off times and available start times from the LP results to those of the initial problem confirmed that all extra delay was eliminated from the schedule in each of the three cases, implying the LP model is a valid representation of the scheduling problem.

IV.4 Strengths and Weaknesses of the LP

While this formulation is guaranteed to provide the optimal schedule for a given cargo flow, any model of a real system will have limitations or weaknesses in addition to its strengths. Beyond the optimality guarantee, other strong points of this method stem from the ease of combining this research with the current procedure at AMC, as well as the additional information provided by the LP solution. The weak points result from the formulation itself, the assumptions made in Chapter I, and the dependence on the cargo generation forecast.

The results of this research can easily be built into the two-step approach proposed for improving the schedule for the channel missions, saving several days of work (using the current "trial and error" method) each time the advance planning process needs to be done by the Command Analysis Group (AMC/XPYR). Since this research relied completely on data provided by AMC/XPYR, all parts were designed to process the data files used by (or created by) STORM, CARGPREP, and CARGOSIM. Any information that was needed for this method (except a feasible cargo flow) was gained from one or more of these files through several user-written FORTRAN programs which pre-processed the data to create the MPS format of the LP formulation. The LP solution provides the final answer required of this step -- the Take-Off times of all the flight legs. These Take-Off times can easily be translated back into the format of 'SCHEDULE.RAW' for entry into CARGOSIM or for use in Step One to re-adjust the cargo flow (i.e., another iteration of the two-step schedule adjustment). Once successfully combined with Step One, this research will therefore provide a time-saving module for refining the schedule of the AMC channel missions without requiring any changes in the existing programs.

The final product of this Step is the solution to the LP problem, so dual variables are readily available. These *shadow prices*, in general, indicate how much the objective function would change for a small change in the related **Take-Off time** or **available start time** when changing one variable at a time. Dual variables could indicate the most beneficial changes in the current cargo flow or in the type of aircraft flying a particular flight. Although this research did not pursue the use of dual variables, the method provides the dual prices as a result of solving the linear program.

These positive aspects of this method are countered by several shortcomings which were beyond the scope of this research. Due to computer limitations, the entire channel system could not be modeled as a single problem using this formulation; while this can be avoided by dividing the full problem into subproblems (four separate theaters with one-week planning horizons) and assuming the subproblems are independent, this assumption may not be acceptable. Not only is there some interaction between the theaters (as noted in Section III.7), but also between the planning periods. Dividing the subproblem for any one theater such that the time horizon is only one week may create problems for modeling the correct number of missions. Since AMC's program, CARGPREP, spreads multiple occurrences of the same mission evenly throughout the month, and since fractions of missions are not modeled by this research, only missions which are flown a multiple of four times in one month will be accurately modeled in the formulation of a one-week time horizon. For example, if a mission is to be flown seven times during the month, CARGPREP will schedule a flight every four days, but the first flight is scheduled arbitrarily; if this first flight is scheduled for the fourth day of the month, the second would not be scheduled until the eighth day. By looking at only one week, the subproblem would be modeled with only one flight of this mission. This problem becomes even more significant for missions that are flown only one, two, or three times in a month -- this method could miss entire missions. Limiting the time horizon was necessary, though, to reduce the problem size and thus the size of the model.

This model tracked aircraft along their individual flights but not <u>between</u> flights, so there is nothing preventing a single aircraft from being assigned to two different flights during overlapping times. While this could be prevented by listing successive flights by an aircraft using consecutively numbered flight legs and then including these flight legs in the set F, the current AMC data does not provide the aircraft identification necessary for this.

Airbases were assumed to have infinite handling capacity (see Section I.5), so nothing prevents every aircraft in the system from landing at the same airbase at the exact same time. While this example is an extreme case, it is possible for a small or very busy airbase to be overwhelmed by multiple demands for unloading, processing, storing, and loading cargo. Also, there are no limits as to the time of day for any activity, which may be an important factor, especially at some of the smaller bases which may have limited time windows for handling cargo and aircraft due to smaller workforces. If an aircraft arrives at such a base in the middle of the night, it might not get serviced until morning, significantly altering its schedule.

While incorporating all cargo requirements that are specified by the cargo flow, this model does not address the "frequency of visit" requirements (see Section I.2) or any passenger requirements (see Section I.5) which comprise a smaller, but still significant, portion of AMC's channel missions. The passenger requirements could be modeled in the same way as cargo, since each passenger has a defined origin and destination; also, AMC currently models every twelve passengers as one pallet load of cargo. The major problem with modeling passenger requirements, though, is trying to determine an accurate forecast. The "frequency of visit" requirements cannot be modeled like cargo because they are not designated by an O-D pair. The complexity involved with modeling these requirements far exceeds the scope of this thesis.

The goal of the LP formulation is to minimize the cumulative weighted time-in-system, but the true, final measure of the cargo's timeliness is comparison to the UMMIPS standards. This research was designed to improve the current schedule by minimizing the delay enroute for a given cargo flow, which means each customer will receive his cargo as early as possible; however, the actual goal of schedulers at AMC's Tanker and Airlift Control Center (TACC) is to deliver the cargo within UMMIPS standards (Berg, 22 Sep 92). Using these standards, the goal of the LP could be to minimize the number of tardy jobs (deliveries that exceed the appropriate UMMIPS standards), perhaps weighted by the size or priority of the cargo.

A final, important limitation of this method stems from the need for a feasible cargo flow, which is based on the cargo generation forecast; this forecast is the cornerstone of the entire procedure, yet it is extremely questionable (Berg, 22 Sep 92). While historic trends do provide some indication of future cargo demand, changing world situations (including base closures and military deployments) add considerable uncertainty. As long as this research uses the same forecasts that were used by STORM, CARGPREP, and the cargo flow procedure of Step One, though, the results should be consistent, and the revised schedule will have the minimum delay enroute for the given cargo flow.

V. Conclusions and Recommendations

V.1 Conclusions

The actual success of modeling the AMC channel cargo distribution system with the LP formulation was documented in Chapter IV. From a theoretical standpoint, given the cargo flow, semi-active scheduling guarantees the minimum delay enroute. From a practical view, this means an aircraft should depart on any particular flight leg as early as physically possible without leaving behind any of its assigned cargo.

Through the use of a linear programming model, this research revised AMC's initial schedule for the channel cargo missions to eliminate any excess delay by minimizing the cumulative, weighted time-in-system for all cargo, according to a given cargo flow. In fact, the revised schedule minimizes <u>any</u> assigned non-negative weighting of the time-in-system, due to the properties of equivalent measures of performance. When combined with Step One of the proposed two-step process for revising AMC's channel mission schedule, this research can be used to improve the current schedule based on Step One's cargo flow.

Currently, this method cannot model the entire AMC channel cargo system due to limitations of computer capacity. To compensate for this, the cargo system was divided into four separate theaters, and the European and Southwest Asia theater was chosen to be formulated for this research because the theater is large and has a considerable number of transshipment requirements. Trying to schedule the missions for one month of this theater created an LP model that was still too large to handle, so the problem was further reduced to a one-week time horizon. This subproblem was solved successfully for three sample cargo flows, and the results indicate significant reductions in the average time-in-system.

By carefully defining the notation and adapting the job-shop formulation, this research devised a method for modeling a limited-size portion of AMC's channel system and minimizing the delay enroute. If future research can improve this method using the recommendations below, this method could become a significant part of AMC's advance planning process.

V.2 Recommendations

The strengths and weaknesses in Section IV.4 indicate the following areas for possible future research: building the process to produce the cargo flow (Step One); combining the two steps into the proposed schedule improvement module; employing relaxation techniques to solve a model of the entire problem; expanding the LP formulation to include constraints related to multiple flights of a single aircraft and to limited capacity or operating hours of the airbases; and improving the cargo generation forecast.

Since the method developed by this research requires a feasible cargo flow prior to improving the current schedule, an obvious first step for future research is to develop a procedure which combines cargo requirements with an initial flight schedule and determines the flow of cargo which minimizes the cargo's delay enroute (i.e., Step One). The results of the cargo flow procedure determine the technological constraints that form the basis of Equations (14) and (15) in this research's method. Currently, the method used for this research employs a simplified, greedy approach to flowing cargo to develop the needed cargo flow information, but the approach does not guarantee to flow all of the cargo or consider the delay enroute. Since the solution to the LP model provides dual variables, one possible avenue of research to design the cargo flow procedure could entail interchange procedures which use the dual variables to indicate approximate benefits from changing the cargo flow.

After Step One is successfully developed, this research can be combined with the cargo flow procedure, producing the proposed two-step approach for improving the schedule for the channel missions. This schedule improvement module can be integrated into the current routine used by the Command Analysis Group (AMC/XPYR) for advance planning purposes. Once successfully combined with Step One, this research will therefore provide a time-saving module for refining the schedule of the AMC channel missions without requiring any changes to AMC's existing programs.

Due to computer limitations, the entire channel system could not be modeled as a single problem using this formulation; however, the subproblems developed in this research are not truly independent. Unless significant advancements are made in computer technology in the near future, future research should consider techniques which might have the ability to solve the scheduling of the entire system as a single problem. A simple algorithm that would iteratively re-schedule flights without violating the cargo flow constraints should be investigated.

The current LP formulation could be expanded to include constraints related to multiple flights of a single aircraft and to limited capacity or operating hours of the airbases. Since this model tracked aircraft along their individual flights but not between flights, there is nothing preventing a single aircraft from being assigned to two different flights during overlapping times. This could easily be prevented by listing the successive flights by an aircraft with consecutively numbered flight legs and then including these flight legs in the set F, but AMC data currently does not provide the aircraft identification necessary for this. If the actual aircraft cannot be individually identified, another method might be to constrain the number of each type of aircraft in use at any given time.

Additional constraints which might add to the realism of this model would involve each base's cargo handling capacity and hours of operation. Airbases were assumed to have infinite cargo handling capacity, so nothing prevents every aircraft in the system from landing at the same airbase at the same time. Also, there are no limits as to the time of day for any activity. Time of day may be an important factor, especially at some of the smaller bases which may have limited time windows for handling cargo and aircraft due to smaller workforces. Future research could investigate ways to incorporate these concerns into the current formulation, perhaps by modeling the capacity of a base as a constrained resource and by developing time window constraints.

Finally, the cargo generation forecast is the basis of the cargo flow which is the basis for this research, but these forecasts "are notorious for their inaccuracies" (Borsi, 11 Apr 92). Research is recommended to investigate the current procedure for developing the forecast, to compare previous forecasts with the actual cargo demand, and to modify the data collection or estimation procedures upon which the forecast is based. While this recommendation does not apply directly to this research, improvements in forecasting the cargo generation for the AMC channel system would have far-reaching effects. AMC's entire advance planning process would benefit greatly.

Appendix A: Airbases in the European Theater

This appendix lists the forty airbases in the European Theater (Europe and Southwest Asia) obtained from a recent AMC study (Robinson, 22 Sep 92). The first column is the number assigned to the airbase in the AMC study, the next is the ICAO code for the airbase. The ICAO code is a four-letter designation used by AMC to identify each airbase.

7 BIKF
8 CYQX
11 DRRN
12 EDAF
13 EDAR
20 EGUN
25 EXXX
28 FTTJ
29 FZAA
30 GLRB
31 GOOY
35 HKNA
37 HSSS
39 KCHS
41 KDOV
43 KGSB
46 KNGU
50 KSBD
51 KSUU
53 KTIK
54 KWRI
55 KXXX
59 LERT
61 LETO
64 LGIR
65 LGSA
69 LICZ
71 LIEO
72 LIPA
73 LIRN
74 LIRP
75 LLBG
77 LPLA
79 LTAG
103 OBBI
104 OEDR
108 OERY
111 OJAF
112 OKBK

113 OMFJ

Appendix B: Cargo Generation Forecast for European Theater

This appendix contains the cumulative amounts of the cargo generated during a one-week period, beginning on Friday. Extracted from the 'DEMAND.RAW' file of a recent AMC study (Robinson, 22 Sep 92), this data was used as input data ('DMDEURO.DAT') for the subproblems in this research. The first two columns in the table show the cargo's O-D pair. The remaining columns show the cumulative tonnage of cargo generated by the origin base for each day of the week. Due to its size, only a portion of this file is presented.

EDAF	EGUN	3.60	7.20	10.80	14.39	17.99	21.59	25.19
EDAF	KCHS	0.88	1.76	2.64	3.52	4.40	5.28	6.16
EDAF	KDOV	36.17	72.34	108.51	144.68	180.85	217.02	253.19
EDAF	KSBD	0.28	0.55	0.83	1.11	1.38	1.66	1.94
EDAF	KSUU	1.28	2.57	3.85	5.13	6.42	7.70	8.98
EDAF	KTIK	9.46	18.92	28.38	37.84	47.30	56.76	66.22
EDAF	KWRI	1.65	3.29	4.94	6.59	8.23	9.88	11.53
EDAF	LETO	0.49	0.98	1.47	1.95	2.44	2.93	3.42
EDAF	LGIR	0.62	1.25	1.87	2.49	3.12	3.74	4.36
EDAF	LIPA	1.48	2.95	4.43	5.91	7.39	8.86	10.34
EDAF	LIRN	0.40	0.81	1.21	1.61	2.02	2.42	2.82
EDAF	LTAG	5.81	11.62	17.42	23.23	29.04	34.85	40.66
EDAF	OEDR	5.16	10.31	15.47	20.62	25.78	30.93	36.09
EDAF	OEJD	0.61	1.23	1.84	2.45	3.07	3.68	4.29
EDAF	OERY	3.31	6.63	9.94	13.25	16.57	19.88	23.19
EDAR	EGUN	3.26	6.52	9.78	13.04	16.30	19.56	22.82
EDAR	KCHS	1.30	2.59	3.89	5.19	6.48	7.78	9.08
EDAR	KDOV	16.19	32.37	48.56	64.75	80.93	97.12	113.31
EDAR	KNGU	0.24	0.48	0.72	0.96	1.20	1.44	1.68
EDAR	KSUU	1.68	3.35	5.03	6.71	8.38	10.06	11.74
EDAR	KTIK	3.02	6.04	9.06	12.08	15.10	18.12	21.14
EDAR	KWRI	1.37	2.73	4.10	5.47	6.83	8.20	9.57

.
.
.

OEDR	KCHS	0.01	0.02	0.03	0.04	0.05	0.06	0.07
OEDR	KDOV	0.37	0.75	1.12	1.49	1.87	2.24	2.61
OEDR	KSBD	0.09	0.19	0.28	0.37	0.47	0.56	0.65
OEDR	KTIK	0.01	0.03	0.04	0.05	0.07	0.08	0.09
OERY	EDAF	0.18	0.35	0.53	0.71	0.88	1.06	1.24
OERY	EDAR	0.59	1.17	1.76	2.35	2.93	3.52	4.11
OERY	KDOV	0.58	1.16	1.74	2.32	2.90	3.48	4.06
OERY	KTIK	0.35	0.70	1.05	1.40	1.75	2.10	2.45
OJAF	KDOV	0.00	0.01	0.01	0.01	0.02	0.02	0.02
OMFJ	KNGU	0.13	0.25	0.38	0.51	0.63	0.76	0.89
OMFJ	LICZ	0.36	0.71	1.07	1.43	1.78	2.14	2.50
OMFJ	OBBI	1.13	2.26	3.39	4.52	5.65	6.78	7.91

Appendix C: Distinct Cargo Pieces for European Theater

This appendix contains an extract of the result ('DMDEURO.OUT') of processing the 'DMDEURO.DAT' file through a user-written FORTRAN program, 'DEMAND.FOR' (Appendix O). The file lists the origin and destination bases, followed by seven sets of columns, where each set lists the cumulative quantity to date and then the size of the small piece and the number of large (5-ton) pieces generated that day.

```
EDAF EGUN   3.60   3.60 0   7.20   3.60 0  10.80   3.60 0  14.39   3.59 0
         17.99   3.60 0  21.59   3.60 0  25.19   3.60 0
EDAF KCHS   0.88   0.00 0   1.76   1.76 0   2.64   0.00 0   3.52   1.76 0
          4.40   0.00 0   5.28   1.76 0   6.16   0.00 0
EDAF KDOV  36.17   1.17 7  72.34   1.17 7 108.51   1.17 7 144.68   1.17 7
        180.85   1.17 7 217.02   1.17 7 253.19   1.17 7
EDAF KSBD   0.28   0.00 0   0.55   0.00 0   0.83   0.83 0   1.11   0.00 0
          1.38   0.00 0   1.66   0.83 0   1.94   0.00 0
EDAF KSUU   1.28   1.28 0   2.57   1.29 0   3.85   1.28 0   5.13   1.28 0
          6.42   1.29 0   7.70   1.28 0   8.98   1.28 0
EDAF KTIK   9.46   4.46 1  18.92   4.46 1  28.38   4.46 1  37.84   4.46 1
         47.30   4.46 1  56.76   4.46 1  66.22   4.46 1
EDAF KWRI   1.65   1.65 0   3.29   1.64 0   4.94   1.65 0   6.59   1.65 0
          8.23   1.64 0   9.88   1.65 0  11.53   1.65 0
EDAF LETO   0.49   0.00 0   0.98   0.00 0   1.47   1.47 0   1.95   0.00 0
          2.44   0.00 0   2.93   1.46 0   3.42   0.00 0
EDAF LGIR   0.62   0.00 0   1.25   1.25 0   1.87   0.00 0   2.49   1.24 0
          3.12   0.00 0   3.74   1.25 0   4.36   0.00 0
EDAF LIPA   1.48   1.48 0   2.95   1.47 0   4.43   1.48 0   5.91   1.48 0
          7.39   1.48 0   8.86   1.47 0  10.34   1.48 0
 .  .  .  .  .  .
 .  .  .  .  .  .
 .  .  .  .  .  .
OERY EDAF   0.18   0.00 0   0.35   0.00 0   0.53   0.53 0   0.71   0.00 0
          0.88   0.00 0   1.06   0.53 0   1.24   0.00 0
OERY EDAR   0.59   0.00 0   1.17   1.17 0   1.76   0.00 0   2.35   1.18 0
          2.93   0.00 0   3.52   1.17 0   4.11   0.00 0
OERY KDOV   0.58   0.00 0   1.16   1.16 0   1.74   0.00 0   2.32   1.16 0
          2.90   0.00 0   3.48   1.16 0   4.06   0.00 0
OERY KTIK   0.35   0.00 0   0.70   0.00 0   1.05   1.05 0   1.40   0.00 0
          1.75   0.00 0   2.10   1.05 0   2.45   0.00 0
OJAF KDOV   0.00   0.00 0   0.01   0.00 0   0.01   0.01 0   0.01   0.00 0
          0.02   0.00 0   0.02   0.01 0   0.02   0.00 0
OMFJ KNGU   0.13   0.00 0   0.25   0.00 0   0.38   0.38 0   0.51   0.00 0
          0.63   0.00 0   0.76   0.38 0   0.89   0.00 0
OMFJ LICZ   0.36   0.00 0   0.71   0.00 0   1.07   1.07 0   1.43   0.00 0
          1.78   0.00 0   2.14   1.07 0   2.50   0.00 0
OMFJ OBBI   1.13   1.13 0   2.26   1.13 0   3.39   1.13 0   4.52   1.13 0
          5.65   1.13 0   6.78   1.13 0   7.91   1.13 0
TOTAL # OF PIECES NEEDING TRANSPORT =  883
```

Appendix D: Routes for European Theater

This appendix contains the routes used as input data ('RTEEURO.DAT') for the subproblems in this research. The data was obtained from the 'ROUTE.DAT' and the 'PLANES.OUT' files of a recent AMC study (Robinson, 22 Sep 92). The first column contains the route number. The subsequent columns outline the specific route using the four-letter ICAO code for each stop and a code number to designate the reason for the stop. The code number is cross-referenced with 'JET.DAT' to determine the required ground times.

3 EXXX1 KTIK4 CYQX4 EDAR4 EXXX9
56 KSUU1 KTIK4 KDOV6 EDAF6 KDOV6 KTIK4 KSUU9
58 KSUU1 KTIK4 KDOV6 EDAR6 KDOV6 KTIK4 KSUU9
59 KSUU1 KTIK4 KDOV6 EGUN6 EDAR4 EDAF6 KCHS6 KTIK4 KSUU9
137 KXXX1 KTIK4 EDAF4 KDOV4 KTIK4 KXXX9
180 KDOV1 EDAF6 KDOV9
181 KDOV1 EDAR6 KDOV9
186 KCHS1 EGUN6 KCHS9
191 KGSB1 KNGU4 LERT6 KNGU4 KGSB9
196 KCHS1 KNGU4 LPLA6 GOOY6 GLRB4 FZAA6 FTTJ4 FZAA6 GOOY4 LPLA6 KNGU4 KCHS9
197 KCHS1 LPLA6 GOOY6 GLRB4 FZAA6 DRRN4 GOOY6 LPLA6 KCHS9
200 KDOV1 EDAR6 OJAF6 EDAR6 KDOV9
202 KCHS1 KNGU4 BIKF6 EGUN4 KCHS9
203 KDOV1 KCHS4 KNGU4 BIKF6 EGUN4 KDOV9
205 KWRI1 KNGU4 LPLA6 LERT4 LIRN6 LICZ4 LERT6 KNGU4 KWRI9
214 EXXX1 KDOV4 EDAF4 EXXX9
215 EXXX1 KDOV4 EDAR4 EXXX9
216 KCHS1 KNGU4 LERT6 LICZ4 OBBI4 OMFJ6 OBBI4 LICZ6 LERT4 LPLA6 KNGU4 KCHS9
224 KDOV1 EDAF6 OEDR4 EDAF6 KDOV9
225 KSUU1 KTIK4 KWRI6 LPLA4 EDAF6 KWRI6 KTIK4 KSUU9
230 EDAF1 LETO4 LIPA6 EDAR4 EGUN4 EDAF9
231 EDAF1 EGUN4 EDAR6 LIPA4 LETO4 EDAF9
235 EDAF1 OKBK4 OEDR6 OERY4 EDAF9
237 EDAF1 LTAG4 EDAF9
239 EDAR1 LTAG4 EDAR9
241 KDOV1 LETO6 KDOV9
242 KWRI1 LPLA6 KWRI9
248 EGUN1 EDAR4 LETO6 EDAR4 EGUN9
249 EGUN1 EDAR4 LIRP4 LIPA6 LETO4 EDAR4 EGUN9
251 EGUN1 EDAF4 LIPA6 LGIR4 LCRA4 LTAG6 LCRA4 LGIR4 LIPA6 EDAF4 EGUN9
252 KDOV1 EDAR4 LTAG4 EDAR4 KDOV9
253 KDOV1 LETO4 LICZ6 LTAG4 LICZ6 LETO4 KDOV9
255 KDOV1 KNGU4 LERT6 OBBI4 LICZ6 LERT6 KNGU4 KDOV9
259 KCHS1 KNGU4 LERT6 LIRN4 LICZ6 LIRN4 LERT6 KNGU4 KCHS9
260 KCHS1 KNGU4 LERT6 LIRN4 LERT6 KNGU4 KCHS9

262 EDAF1 EGUN4 EDAR4 LIPA4 LETO4 EDAF4 LTAG6 EDAF4 LETO4 LIPA4 EDAR4 EGUN4 EDAF9
264 EDAF1 LIRN4 LICZ4 LERT6 LICZ4 LIRN4 EDAF9
265 KCHS1 KNGU4 LERT6 LIRN4 LICZ4 OBBI6 OMFJ4 OBBI4 LICZ6 LIRN4 LERT6 LPLA4 KNGU4 KCHS9
266 EDAF1 LIRN4 LICZ4 LIRN4 EDAF9
267 KCHS1 EGUN6 KCHS9
268 KNGU1 CYQX4 LERT6 LICZ4 LERT4 KNGU9
269 KDOV1 EDAF4 OERY6 EDAF4 KDOV9
270 KWRI1 LPLA4 EDAR6 LPLA4 KWRI9
271 EDAF1 OEDR6 EDAF9
274 LIRN1 LGSA7 LIRN9
275 LIRN1 LIEO7 LIRN9
292 EDAF1 EDAR4 EDAF9
293 KDOV1 EDAR4 LLBG4 EDAR4 KDOV9
294 KNGU1 LETO4 LICZ4 HSSS4 HKNA4 LICZ4 LPLA4 KNGU9

Appendix E: Initial Flight Schedule

This appendix contains an extract of the information used as the initial flight schedule for the subproblems in this research. The data was obtained from the 'SCHEDULE.RAW' file of a recent AMC study (Robinson, 22 Sep 92). The first column contains the route number, the second column contains the aircraft type selected for that route, and the third column contains the day that the aircraft departs its origin base (decimals indicate the fraction of day that the aircraft departs on the initial flight leg -- successive flight legs begin immediately after the required ground time).

19	C005	0.1
19	C005	15.1
23	C005	1.2
37	C005	2.3
56	C005	3.4
58	C005	4.5
58	C005	12.0
58	C005	19.5
58	C005	27.0
60	C005	5.6
	.	
	.	
	.	
	.	
	.	
252	KC10	12.5
252	KC10	14.8
252	KC10	17.1
252	KC10	19.5
252	KC10	21.8
252	KC10	24.1
252	KC10	26.4
252	KC10	28.7
252	KC10	1.0
253	KC10	4.4

Appendix F: Flight Times Between Bases

This appendix contains an extract of the flight times between airbases used as input data for the subproblem in this research. The data was obtained from the 'FLY.DAT' file of a recent AMC study (Robinson, 22 Sep 92). The first two columns contain the ICAO codes for the starting and ending airbases of a flight leg, and the remaining columns contain the flight times (in hours) between the two airbases for the various aircraft types. The fourth column contains the flight times for a C141 aircraft. AMC actually only uses the fourth column in the table to calculate flight times for the other aircraft types by using a multiplication factor in the 'JET.DAT' file of the recent AMC study.

ABAS	ASRI	2.7	2.7	2.7	2.7	2.7	2.7	2.7
APLM	ASRI	4.7	4.7	4.7	4.7	4.7	4.7	4.7
APWR	ASRI	1.8	1.8	1.8	1.8	1.8	1.8	1.8
ASRI	ABAS	3.0	3.0	3.0	3.0	3.0	3.0	3.0
ASRI	APLM	5.8	5.8	5.8	5.8	5.8	5.8	5.8
ASRI	APWR	2.2	2.2	2.2	2.2	2.2	2.2	2.2
ASRI	NSTU	5.5	5.5	5.5	5.5	5.5	5.5	5.5
ASRI	NZCH	3.0	3.0	3.0	3.0	3.0	3.0	3.0
BGSF	BGTL	1.8	1.8	1.8	1.8	1.8	1.8	1.8
BGSF	CYYR	2.7	2.7	2.7	2.7	2.7	2.7	2.7
.								
.								
.								
KSUU	KRIV	1.9	1.9	1.9	1.9	1.9	1.9	1.9
KSUU	PADK	6.5	6.5	6.5	6.5	6.5	6.5	6.5
LERT	OBBI	7.0	7.0	7.0	7.0	7.0	7.0	7.0
PGUA	RJTY	4.2	4.2	4.2	4.2	4.2	4.2	4.2
PHIK	PWAK	5.4	5.4	5.4	5.4	5.4	5.4	5.4
PHIK	RODN	10.2	10.2	10.2	10.2	10.2	10.2	10.2
RODN	WSAP	5.6	5.6	5.6	5.6	5.6	5.6	5.6
RPMB	WIIH	4.2	4.2	4.2	4.2	4.2	4.2	4.2
WIIH	RPMB	4.4	4.4	4.4	4.4	4.4	4.4	4.4
WSAP	RODN	5.1	5.1	5.1	5.1	5.1	5.1	5.1

Appendix G: Flight Legs for European Theater

This appendix contains the numbering system for the 377 distinct flight legs that make up the 81 distinct flights flown in one week of the European theater. The first column contains the distinct number assigned to each flight, and the remaining columns contain the distinct numbers assigned to each leg of the flight.

1	1	2	3	4							
2	5	6	7	8	9	10					
3	11	12	13	14	15	16					
4	17	18	19	20	21	22	23	24			
5	25	26	27	28	29	30	31	32			
6	33	34	35	36	37	38	39	40			
7	41	42	43	44	45						
8	46	47	48	49	50						
9	51	52									
10	53	54									
11	55	56									
12	57	58									
13	59	60									
14	61	62									
15	63	64									
16	65	66									
17	67	68	69	70							
18	71	72	73	74	75	76	77	78			
19	79	80	81	82							
20	83	84	85	86							
21	87	88	89	90	91						
22	92	93	94	95	96	97	98	99			
23	100	101	102								
24	103	104	105								
25	106	107	108								
26	109	110	111	112	113	114	115	116	117	118	119
27	120	121	122	123	124	125	126	127	128	129	130
28	131	132	133	134	135	136	137	138	139	140	141
29	142	143	144	145							
30	146	147	148	149	150	151	152				
31	153	154	155	156	157						
32	158	159	160	161	162						
33	163	164	165	166	167						
34	168	169	170	171	172						
35	173	174	175	176	177						
36	178	179	180	181							
37	182	183	184	185							
38	186	187									
39	188	189									
40	190	191									
41	192	193									
42	194	195									

43	196	197											
44	198	199	200	201									
45	202	203	204	205									
46	206	207	208	209	210	211							
47	212	213	214	215	216	217	218	219	220	221			
48	222	223	224	225	226	227	228	229	230	231			
49	232	233	234	235									
50	236	237	238	239									
51	240	241	242	243									
52	244	245	246	247	248	249							
53	250	251	252	253	254	255							
54	256	257	258	259	260	261	262						
55	263	264	265	266	267	268	269	270					
56	271	272	273	274	275	276							
57	277	278	279	280	281	282							
58	283	284	285	286	287	288							
59	289	290	291	292	293	294	295	296	297	298	299	300	
60	301	302	303	304	305	306							
61	307	308	309	310	311	312	313	314	315	316	317	318	319
62	320	321	322	323									
63	324	325	326	327									
64	328	329											
65	330	331											
66	332	333	334	335	336								
67	337	338	339	340									
68	341	342	343	344									
69	345	346											
70	347	348											
71	349	350											
72	351	352											
73	353	354											
74	355	356											
75	357	358											
76	359	360											
77	361	362											
78	363	364											
79	365	366											
80	367	368	369	370									
81	371	372	373	374	375	376	377						

Appendix H: Transshipment Data for European Theater

This appendix contains the 48 transshipment combinations available within the European theater. Each combination details the initial route number; the origin, transshipment, and destination bases; the possible follow-on routes; and finally the route number(s) of any direct route(s), when applicable. This data ('TRNSEURO.DAT') is used by the user-written program 'CARGFLOW.FOR' (see Appendix Q) to determine an initial cargo flow.

200	EDAR	KDOV	KNGU	203 255	0
59	EDAR	EDAF	LGIR	251	0
292	EDAR	EDAF	LGIR	251	0
59	EDAR	EDAF	LICZ	264 266	0
59	EDAR	EDAF	LIRN	264 266	0
59	EDAR	EDAF	OEDR	224 235 271	0
230	EDAR	EDAF	OEDR	224 235 271	0
202	EGUN	KCHS	KNGU	196 202 203 216 259 260 265	0
231	EGUN	EDAR	LPLA	270	0
231	EGUN	EDAR	LTAG	239 252 262	251 262
260	KCHS	LIRN	EDAF	264 266	0
186	KCHS	EGUN	LETO	231 248 249 262	0
59	KDOV	EGUN	LGIR	251	0
181	KDOV	EDAR	LIPA	231 249 262	0
241	KDOV	LETO	LIPA	230 262	0
269	KDOV	EDAF	LIPA	230 231 251 262	0
56	KDOV	EDAF	OEDR	224 235 271	224
180	KDOV	EDAF	OEDR	224 235 271	224
202	KNGU	EGUN	LIPA	231 249 251 262	0
203	KNGU	EGUN	LIPA	231 249 251 262	0
60	KSUU	KDOV	OBBI	255	0
56	KSUU	KDOV	OEDR	224	0
60	KTIK	KDOV	LETO	241 253	0
137	KTIK	EDAF	LGIR	251	0
137	KTIK	EDAF	LIPA	230 231 251 262	0
3	KTIK	EDAR	LTAG	239 252 262	0
137	KTIK	EDAF	LTAG	237 251 262	0
137	KTIK	EDAF	OEDR	224 235 271	0
137	KTIK	EDAF	OERY	235 269	0
264	LERT	EDAF	OEDR	224 235 271	0
249	LETO	EGUN	KDOV	203	241 253
231	LETO	EDAF	KTIK	56 59 137 225	0
230	LETO	EDAR	KWRI	270	0
231	LETO	EDAF	KWRI	225	0
294	LETO	LPLA	KWRI	225 242 270	0
231	LETO	EDAF	LERT	264	0
231	LETO	EDAF	LGIR	251	0
231	LETO	EDAF	LIRN	264 266	0
253	LETO	LICZ	OBBI	216 265	0

266 LICZ EDAF KSUU	56 59 225	0
251 LIPA EDAF KDOV	56 137 180 224 269	0
230 LIPA EDAF KWRI	225	0
225 LPLA EDAF LETO	230 231 262	0
237 LTAG EDAF EGUN	230 231 251 262	251 262
237 LTAG EDAF LIRN	264 266	0
252 LTAG EDAR LLBG	293	0
235 OERY EDAF EDAR	230 231 262	0
269 OERY KDOV KTIK	56 58 59 137	0

Appendix I: Detailed Flight Schedule for European Theater

This appendix contains an extract of the results ('SCHEDULD.PRN') of a user-written program ('SCHEDULD.FOR') that combines the initial flight schedule and aircraft capacity based on the aircraft type ('SCHEDULE.RAW'), the routes in the European theater ('RTEEURO.DAT'), the flight times between bases ('FLY.DAT'), and the ground times based on the stop codes of the routes ('JET.DAT'). This data is formatted as follows: the first column contains a distinct, user-assigned flight number; the second column lists the route number; the third column tracks the number of times (including this one) the route has been flown up to this point; the fouth column provides a count of the number of bases on the route; the fifth column displays the aircraft capacity (in tons); and the remaining columns depend on the number of bases on the route, with a format of departure base and time, followed by arrival time and base (which, of course, is also the next departure base if the route continues).

```
FLT  RTE #  B  C
            BASE DEP   ARR  BASE  DEP   ARR  BASE...
 1   3  1  5 25
            EXXX  79.2  79.2 KTIK  82.2  86.9 CYQX  89.9  96.1
            EDAR  99.1  99.1 EXXX
 2  56  1  7 50
            KSUU  81.6  84.5 KTIK  88.8  91.6 KDOV 109.8 117.8
            EDAF 136.0 145.6 KDOV 163.9 167.1 KTIK 171.3 174.7
            KSUU
 3  58  1  7 50
            KSUU 108.0 110.9 KTIK 115.2 118.0 KDOV 136.2 144.2
            EDAR 162.4 171.6 KDOV 189.9 193.1 KTIK 197.3 200.7
            KSUU
 4  59  1  9 18
            KSUU   0.0   3.0 KTIK   6.2   9.1 KDOV  26.4  33.5
            EGUN  50.8  52.2 EDAR  55.4  55.5 EDAF  72.8  83.3
            KCHS 100.6 103.4 KTIK 106.7 110.2 KSUU
 5  59  2  9 18
            KSUU  72.0  75.0 KTIK  78.2  81.2 KDOV  98.4 105.5
            EGUN 122.8 124.2 EDAR 127.4 127.5 EDAF 144.8 155.4
            KCHS 172.6 175.4 KTIK 178.7 182.2 KSUU
 . . . . .
 . . . . .
 . . . . .
80 293  1  5 50
            KDOV   0.0   8.0 EDAR  12.2  16.4 LLBG  20.6  25.9
            EDAR  30.1  39.3 KDOV
81 294  1  8 18
            KNGU 158.4 166.5 LETO 169.8 172.2 LICZ 175.5 180.4
            HSSS 183.6 186.5 HKNA 189.8 197.4 LICZ 200.6 206.1
            LPLA 209.4 216.0 KNGU
```

Appendix J: Detailed Cargo Listing for European Theater

This appendix contains an extract of one of the output files ('CARGPICS.OUT') from a user-written program ('CARGFLOW.FOR') that re-formats the file containing the 883 distinct cargo pieces for one week of the European theater ('DMDEURO.OUT'). This new format assigns a distinct number to each piece, as well as listing each piece's origin and destination bases, time of creation (in hours), and size (in tons).

1	EDAF	EGUN	0	3.60
2	EDAF	KDOV	0	1.17
3	EDAF	KDOV	0	5.00
4	EDAF	KDOV	0	5.00
5	EDAF	KDOV	0	5.00
6	EDAF	KDOV	0	5.00
7	EDAF	KDOV	0	5.00
8	EDAF	KDOV	0	5.00
9	EDAF	KDOV	0	5.00
10	EDAF	KSUU	0	1.28
11	EDAF	KTIK	0	4.46
12	EDAF	KTIK	0	5.00
13	EDAF	KWRI	0	1.65
14	EDAF	LIPA	0	1.48
15	EDAF	LTAG	0	0.81
16	EDAF	LTAG	0	5.00
17	EDAF	OEDR	0	0.16
18	EDAF	OEDR	0	5.00
19	EDAF	OERY	0	3.31
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
865	LETO	KWRI	144	1.16
866	LETO	LIPA	144	1.71
867	LETO	LIPA	144	5.00
868	LETO	LTAG	144	3.99
869	LICZ	KNGU	144	4.28
870	LICZ	LERT	144	1.85
871	LICZ	LIRN	144	1.49
872	LICZ	LLBG	144	2.25
873	LICZ	LTAG	144	1.24
874	LIPA	EDAF	144	1.16
875	LIPA	KWRI	144	1.58
876	LPLA	KWRI	144	3.13
877	LTAG	EDAF	144	3.38
878	LTAG	EDAR	144	2.39
879	LTAG	EGUN	144	1.95
880	LTAG	KDOV	144	2.57
881	LTAG	KDOV	144	5.00
882	LTAG	LLBG	144	1.07
883	OMFJ	OBBI	144	1.13

Appendix K: Detailed Cargo Flow for European Theater

This appendix contains an extract of the three different cargo flows ('CARGLEGS.OUT') assigned by a user-written program ('CARGFLOW.FOR'). The first column assigns a revised number to each piece of cargo that was successfully flowed through the system by the FORTRAN program. The second column shows each piece's original number of the 883 pieces in 'CARGPICS.OUT'. The third and fourth columns list each piece's time of creation (in hours) and size (in tons). The fifth column counts the number of legs being used to transport a piece. The sixth column provides the leg number assigned to transport the piece, in order. The next four columns detail the assigned leg number's initial schedule by showing the departure and arrival times and bases. The eleventh column tracks each piece's current time in system, based on the completion of the current flight leg, including the ground processing time which is listed in the last column.

K.1 Detailed Cargo Flow #1

New #	Old #	Mark Time	Size (Wt)	Job #	Leg #	Dep. Time	Arr. Time	Dep. Base	Arr. Base	Time in Sys	Proc Time
1	1	0	3.60	1	153	28.80	31.40	EDAF	LETO	34.70	5.90
1	1	0	3.60	2	154	34.70	36.90	LETO	LIPA	54.10	19.40
1	1	0	3.60	3	155	54.10	55.90	LIPA	EDAR	59.20	5.10
1	1	0	3.60	4	156	59.20	60.70	EDAR	EGUN	64.70	5.50
2	2	24	3.60	1	153	28.80	31.40	EDAF	LETO	10.70	5.90
2	2	24	3.60	2	154	34.70	36.90	LETO	LIPA	30.10	19.40
2	2	24	3.60	3	155	54.10	55.90	LIPA	EDAR	35.20	5.10
2	2	24	3.60	4	156	59.20	60.70	EDAR	EGUN	40.70	5.50
3	3	48	3.60	1	158	79.20	81.80	EDAF	LETO	37.00	5.80
3	3	48	3.60	2	159	85.00	87.20	LETO	LIPA	56.50	19.50
3	3	48	3.60	3	160	104.50	106.30	LIPA	EDAR	61.50	5.00
3	3	48	3.60	4	161	109.50	111.00	EDAR	EGUN	67.00	5.50
4	4	72	3.59	1	158	79.20	81.80	EDAF	LETO	13.00	5.80
4	4	72	3.59	2	159	85.00	87.20	LETO	LIPA	32.50	19.50
4	4	72	3.59	3	160	104.50	106.30	LIPA	EDAR	37.50	5.00
4	4	72	3.59	4	161	109.50	111.00	EDAR	EGUN	43.00	5.50
.	.	.	.	.							
.	.	.	.	.							
.	.	.	.	.							
602	876	120	1.07	1	125	148.60	149.30	OMFJ	OBBI	32.60	4.00
602	876	120	1.07	2	126	152.60	158.80	OBBI	LICZ	42.80	10.20
603	877	0	1.13	1	114	76.70	77.40	OMFJ	OBBI	81.40	4.70
604	878	24	1.13	1	114	76.70	77.40	OMFJ	OBBI	57.40	4.70
605	879	48	1.13	1	114	76.70	77.40	OMFJ	OBBI	33.40	4.70
606	880	72	1.13	1	114	76.70	77.40	OMFJ	OBBI	9.40	4.70
607	881	96	1.13	1	125	148.60	149.30	OMFJ	OBBI	57.30	4.70
608	882	120	1.13	1	125	148.60	149.30	OMFJ	OBBI	33.30	4.70
609	883	144	1.13	1	125	148.60	149.30	OMFJ	OBBI	9.30	4.70

K.2 Detailed Cargo Flow #2

New #	Old #	Mark Time	Size (Wt)	job #	Leg #	Dep. Time	Arr. Time	Dep. Base	Arr. Base	Time in Sys	Proc Time
1	883	144	1.13	1	125	148.60	149.30	OMFJ	OBBI	9.30	4.70
2	882	120	1.13	1	125	148.60	149.30	OMFJ	OBBI	33.30	4.70
3	881	96	1.13	1	125	148.60	149.30	OMFJ	OBBI	57.30	4.70
4	880	72	1.13	1	114	76.70	77.40	OMFJ	OBBI	9.40	4.70
5	879	48	1.13	1	114	76.70	77.40	OMFJ	OBBI	33.40	4.70
6	878	24	1.13	1	114	76.70	77.40	OMFJ	OBBI	57.40	4.70
7	877	0	1.13	1	114	76.70	77.40	OMFJ	OBBI	81.40	4.70
8	876	120	1.07	1	125	148.60	149.30	OMFJ	OBBI	32.60	4.00
8	876	120	1.07	2	126	152.60	158.80	OBBI	LICZ	42.80	10.20
9	875	48	1.07	1	114	76.70	77.40	OMFJ	OBBI	32.60	3.90
9	875	48	1.07	2	115	80.60	86.80	OBBI	LICZ	42.80	10.20
10	874	120	0.38	1	125	148.60	149.30	OMFJ	OBBI	32.60	4.00
10	874	120	0.38	2	126	152.60	158.80	OBBI	LICZ	56.00	23.40
10	874	120	0.38	3	127	176.00	179.20	LICZ	LERT	62.50	6.50
10	874	120	0.38	4	128	182.50	185.50	LERT	LPLA	82.70	20.20
10	874	120	0.38	5	129	202.70	209.30	LPLA	KNGU	93.30	10.60
.	.	.	.	.	.						
.	.	.	.	.	.						
.	.	.	.	.	.						
646	23	24	5.00	1	52	69.10	78.10	EDAF	KDOV	58.10	13.00
647	22	24	5.00	1	52	69.10	78.10	EDAF	KDOV	58.10	13.00
648	21	24	5.00	1	52	69.10	78.10	EDAF	KDOV	58.10	13.00
649	20	24	5.00	1	52	69.10	78.10	EDAF	KDOV	58.10	13.00
650	19	24	1.17	1	43	98.30	107.30	EDAF	KDOV	87.30	13.00
651	18	0	5.00	1	54	105.40	115.00	EDAF	KDOV	119.00	13.60
652	17	0	5.00	1	54	105.40	115.00	EDAF	KDOV	119.00	13.60
653	16	0	5.00	1	54	105.40	115.00	EDAF	KDOV	119.00	13.60
654	15	0	5.00	1	54	105.40	115.00	EDAF	KDOV	119.00	13.60
655	14	0	5.00	1	54	105.40	115.00	EDAF	KDOV	119.00	13.60
656	13	0	5.00	1	54	105.40	115.00	EDAF	KDOV	119.00	13.60
657	12	0	5.00	1	54	105.40	115.00	EDAF	KDOV	119.00	13.60
658	11	0	1.17	1	43	98.30	107.30	EDAF	KDOV	111.30	13.00
659	10	120	1.76	1	38	216.80	227.40	EDAF	KCHS	111.40	14.60
660	9	72	1.76	1	38	216.80	227.40	EDAF	KCHS	159.40	14.60
661	8	24	1.76	1	38	216.80	227.40	EDAF	KCHS	207.40	14.60
662	5	96	3.60	1	173	105.60	107.10	EDAF	EGUN	15.10	5.50
663	4	72	3.59	1	173	105.60	107.10	EDAF	EGUN	39.10	5.50
664	3	48	3.60	1	173	105.60	107.10	EDAF	EGUN	63.10	5.50
665	2	24	3.60	1	168	33.60	35.10	EDAF	EGUN	15.10	5.50
666	1	0	3.60	1	168	33.60	35.10	EDAF	EGUN	39.10	5.50

K.3 Detailed Cargo Flow #3

New #	Old #	Mark Time	Size (Wt)	job #	Leg #	Dep. Time	Arr. Time	Dep. Base	Arr. Base	Time in Sys	Proc Time
1	1	0	3.60	1	153	28.80	31.40	EDAF	LETO	34.70	5.90
1	1	0	3.60	2	154	34.70	36.90	LETO	LIPA	54.10	19.40
1	1	0	3.60	3	155	54.10	55.90	LIPA	EDAR	59.20	5.10
1	1	0	3.60	4	156	59.20	60.70	EDAR	EGUN	64.70	5.50
2	2	0	1.17	1	8	136.00	145.60	EDAF	KDOV	149.60	13.60
3	3	0	5.00	1	8	136.00	145.60	EDAF	KDOV	149.60	13.60
4	4	0	5.00	1	8	136.00	145.60	EDAF	KDOV	149.60	13.60
5	5	0	5.00	1	8	136.00	145.60	EDAF	KDOV	149.60	13.60
6	6	0	5.00	1	8	136.00	145.60	EDAF	KDOV	149.60	13.60
7	7	0	5.00	1	8	136.00	145.60	EDAF	KDOV	149.60	13.60
.	.	.	.	.							
.	.	.	.	.							
.	.	.	.	.							
614	869	144	4.28	1	127	176.00	179.20	LICZ	LERT	38.50	6.50
614	869	144	4.28	2	128	182.50	185.50	LERT	LPLA	58.70	20.20
614	869	144	4.28	3	129	202.70	209.30	LPLA	KNGU	69.30	10.60
615	870	144	1.85	1	134	193.80	199.20	LICZ	OBBI	58.50	8.70
615	870	144	1.85	2	135	202.50	203.40	OBBI	OMFJ	76.60	18.10
615	870	144	1.85	3	136	220.60	221.30	OMFJ	OBBI	80.60	4.00
615	870	144	1.85	4	137	224.60	230.80	OBBI	LICZ	104.00	23.40
615	870	144	1.85	5	138	248.00	251.20	LICZ	LERT	111.20	7.20
616	874	144	1.16	1	230	185.10	188.30	LIPA	EDAF	48.30	7.20
617	877	144	3.38	1	191	166.00	171.20	LTAG	EDAF	31.20	9.20
618	878	144	2.39	1	242	152.60	157.80	LTAG	EDAR	17.80	9.20
619	880	144	2.57	1	242	152.60	157.80	LTAG	EDAR	17.10	8.50
619	880	144	2.57	2	243	161.10	169.80	EDAR	KDOV	29.80	12.70
620	881	144	5.00	1	242	152.60	157.80	LTAG	EDAR	17.10	8.50
620	881	144	5.00	2	243	161.10	169.80	EDAR	KDOV	29.80	12.70
621	883	144	1.13	1	125	148.60	149.30	OMFJ	OBBI	9.30	4.70

Appendix L: Taskings for Flight Legs for European Theater

This appendix contains an extract of the file ('LEGCARGO.OUT') detailing each flight leg's assigned cargo, as determined by a user-written program ('CARGFLOW.FOR'). The first column is the distinct number of the flight leg; the second number provides the flight number (assigned in 'SCHEDULD.PRN'); the third column counts the flight legs for each distinct flight; the fourth column lists the processing time of the leg (in hours); the fifth column contains the count of the distinct cargo pieces to be transported by the leg; the sixth column indicates the required number of constraints relating to the Take-Off times for all flight legs listed prior to the current leg; and the remaining columns contain the revised numbers for the cargo pieces assigned to this leg.

L.1 Taskings for Cargo Flow #1

```
 1   1  1  3.0  0    0
 2   1  2  7.7  7    1 424 425 426 427 428 429 432
 3   1  3  9.2  7    9 424 425 426 427 428 429 432
 4   1  4  4.0  0   17
 5   2  1  7.2  1   18 420
 6   2  2 21.0  1   20 420
 7   2  3 26.2  1   22 420
 8   2  4 27.9 13   24  11  12  13  14  15  16  17  18  19  20  21  27  35
 9   2  5  7.4  2   38 596 597
10   2  6  7.4  0   41
11   3  1  7.2  1   42 422
.  .  .  .  .  .
.  .  .  .  .  .
.  .  .  .  .  .
371  81  1 11.4  3 1374 405 406 407
372  81  2  5.7  4 1378 405 406 407 484
373  81  3  8.1  0 1383
374  81  4  6.2  0 1384
375  81  5 10.8  0 1385
376  81  6  8.8  0 1386
377  81  7 10.6  0 1387
```

Number of Pieces Flowed thru System = 609
Cum. Weighted Time-in-System of pieces = 118213.39
Avg. Weighted Time-in-System per piece = 194.11
Cum. Weight of Pieces Flowed thru System = 1838.77
Avg. Weight of Pieces Flowed thru System = 3.02
Avg. Time-in-System of Pieces thru System = 64.29
Max # of pieces on any one LEG = 15
# of constraints for T.O. Times (incl. non-neg.)= 1388
# of constraints for Cargo Ready Times = 1620
Total # of constraints in LP Formulation = 3008
Total # of variables in LP Formulation = 1997

L.2 Taskings for Cargo Flow #2

```
1   1  1  3.0  0   0
2   1  2  7.7  7   1 206 207 228 229 230 231 232
3   1  3  9.2  7   9 206 207 228 229 230 231 232
4   1  4  4.0  0  17
5   2  1  7.2  1  18 237
6   2  2 21.0  7  20 219 220 221 222 223 224 237
7   2  3 26.2 17  28 219 220 221 222 223 224 237 280 281 282 283 284 285 286 363 372 382
8   2  4 27.9 14  46  78 114 151 584 585 586 587 588 589 590 591 592 597 598
9   2  5  7.4 15  61  13  14 114 151 584 585 586 587 588 589 590 591 592 597 598
10  2  6  7.4  3  77 114 597 598
11  3  1  7.2  1  81 235
12  3  2 21.0  5  83 225 226 227 233 235
13  3  3 26.2 12  89 225 226 227 233 235 310 311 312 313 314 315 318
14  3  4 27.5 18 102 487 488 489 490 491 492 493 494 495 496 497 498 499 500 503 504
 505 506
15  3  5  7.4 14 121 487 488 489 490 491 492 493 494 495 496 497 498 499 500
16  3  6  7.4  7 136 494 495 496 497 498 499 500
17  4  1  6.2  0 144
18  4  2 20.2  0 145
19  4  3 24.4  4 146 334 335 336 340
20  4  4  4.6  5 151 334 335 336 340 420
21  4  5 17.4  8 157 420 457 471 480 481 534 535 536
22  4  6 27.8  9 166 420 534 535 536 593 594 599 600 601
23  4  7  6.1  6 176 420 593 594 599 600 601
24  4  8  7.5  4 183 420 599 600 601
.   .  .  .    .   .
.   .  .  .    .   .
.   .  .  .    .   .
371 81  1 11.4  3 1593 255 256 257
372 81  2  5.7  4 1597 143 255 256 257
373 81  3  8.1  0 1602
374 81  4  6.2  0 1603
375 81  5 10.8  0 1604
376 81  6  8.8  0 1605
377 81  7 10.6  0 1606
```

Number of Pieces Flowed thru System = 666
Cum. Weighted Time-in-System of pieces = 119534.92
Avg. Weighted Time-in-System per piece = 179.48
Cum. Weight of Pieces Flowed thru System = 1996.43
Avg. Weight of Pieces Flowed thru System = 3.00
Avg. Time-in-System of Pieces thru System = 59.87
Max # of pieces on any one LEG = 18
# of constraints for T.O. Times (incl. non-neg.)= 1607
# of constraints for Cargo Ready Times = 1896
Total # of constraints in LP Formulation = 3503
Total # of variables in LP Formulation = 2273

L.3 Taskings for Cargo Flow #3

```
1    1  1  3.0  0    0
2    1  2  7.7  7    1  71  72 152 154 245 365 366
3    1  3  9.2  7    9  71  72 152 154 245 365 366
4    1  4  4.0  0   17
5    2  1  7.2  1   18 243
6    2  2 21.0  2   20  73 243
7    2  3 26.2  3   23  62  73 243
8    2  4 27.9 14   27   2   3   4   5   6   7   8   9  10  11  12 100 108 267
9    2  5  7.4  7   42  10  11  12 108 267 286 575
10   2  6  7.4  3   50  10 108 267
11   3  1  7.2  1   54 244
12   3  2 21.0  1   56 244
13   3  3 26.2  8   58  51  52  53  54  55  56 145 244
14   3  4 27.5 15   67  22  23  24  25  26  27 121 122 123 124 125 126 212 213 217
15   3  5  7.4  5   83  26  27 125 126 217
16   3  6  7.4  3   89  26 125 217
17   4  1  6.2  0   93
18   4  2 20.2  0   94
19   4  3 24.4  1   95  44
20   4  4  4.6  6   97  34  35  36  37  39  44
21   4  5 17.4  6  104  21  34  36  37  39  44
22   4  6 27.8  6  111  21  36  37  39  99 109
23   4  7  6.1  2  118  39 109
24   4  8  7.5  0  121
.    .  .   .   .   .
.    .  .   .   .   .
.    .  .   .   .   .
371 81  1 11.4  3 1419 538 539 540
372 81  2  5.7  4 1423 538 539 540 552
373 81  3  8.1  0 1428
374 81  4  6.2  0 1429
375 81  5 10.8  0 1430
376 81  6  8.8  0 1431
377 81  7 10.6  0 1432
```

Number of Pieces Flowed thru System = 621
Cum. Weighted Time-in-System of pieces = 131121.27
Avg. Weighted Time-in-System per piece = 211.15
Cum. Weight of Pieces Flowed thru System = 1857.32
Avg. Weight of Pieces Flowed thru System = 2.99
Avg. Time-in-System of Pieces thru System = 70.60
Max # of pieces on any one LEG = 16
# of constraints for T.O. Times (incl. non-neg.)= 1433
# of constraints for Cargo Ready Times = 1677
Total # of constraints in LP Formulation = 3110
Total # of variables in LP Formulation = 2054

Appendix M: LP Solution for Scheduling of European Theater

This appendix contains an extract of the solution ('SCHEDRUN.SOL') of the linear programming model of scheduling the flight legs for one week of the European theater. The LP model was solved by MINOS after being converted into MPS format by a user-written program ('SCHEDMPS.FOR') and by using a specification file ('SCHMINOS.SPC') as follows:

```
BEGIN
MINIMIZE
ROWS 3115
COLUMNS 2060
FACTORIZATION FREQUENCY 10
MPS FILE 24
BOUNDS NONE
OBJECTIVE OBJ
END
```

The solution presents the value of the objective function and then the values for the decision variables under the column labeled "activity". The variables are named in a similar fashion to the LP formulation in the research, with *ti,j* representing the available start time of cargo piece *j* on flight leg *i* and *TOi* representing the optimal Take-Off time of flight leg *i* for the given cargo flow.

```
        M I N O S   ---   VERSION 5.0   MAY 1985
        = = = = =

SECTION 1 - ROWS

 NUMBER ...ROW.. STATE ...ACTIVITY... SLACK ACTIVITY  ..I

  2055 OBJ       BS    97101.15700   -97101.15700    1

SECTION 2 - COLUMNS

 NUMBER .COLUMN. STATE ...ACTIVITY... .OBJ GRADIENT.   M+J

     1 t153,  1 D BS     0.00000      -3.60000   3112
     2 t154,  1   BS     5.90000       0.00000   3113
     3 t155,  1   BS    25.30000       0.00000   3114
     4 t156,  1   BS    53.10000       0.00000   3115
     5 tEND,  1   BS    58.60000       3.60000   3116
     6 t  8,  2 D BS     0.00000      -1.17000   3117
     7 tEND,  2   BS   116.00000       1.17000   3118
     8 t  8,  3 D BS     0.00000      -5.00000   3119
     9 tEND,  3   BS   116.00000       5.00000   3120
    10 t  8,  4 D BS     0.00000      -5.00000   3121
    11 tEND,  4   BS   116.00000       5.00000   3122
```

12	t 8, 5 D	BS	0.00000	-5.00000	3123
13	tEND, 5	BS	116.00000	5.00000	3124
14	t 8, 6 D	BS	0.00000	-5.00000	3125
15	tEND, 6	BS	116.00000	5.00000	3126
16	t 8, 7 D	BS	0.00000	-5.00000	3127
17	tEND, 7	BS	116.00000	5.00000	3128
18	t 8, 8 D	BS	0.00000	-5.00000	3129
19	tEND, 8	BS	116.00000	5.00000	3130
20	t 8, 9 D	BS	0.00000	-5.00000	3131
21	tEND, 9	BS	116.00000	5.00000	3132
22	t 8, 10 D	BS	0.00000	-1.28000	3133
23	t 9, 10	BS	130.30000	0.00000	3134
24	t 10, 10	BS	160.00000	0.00000	3135
25	tEND, 10	BS	167.40000	1.28000	3136
.	.	.	.		
.	.	.	.		
.	.	.	.		
1673	t242,620	BS	144.00000	-5.00000	4784
1674	t243,620	BS	159.90000	0.00000	4785
1675	tEND,620	BS	172.60000	5.00000	4786
1676	t125,621	BS	144.00000	-1.13000	4787
1677	tEND,621	BS	151.50000	1.13000	4788
1678	TO 1 D	BS	0.00000	0.00000	4789
1679	TO 2	BS	72.00000	0.00000	4790
1680	TO 3	BS	79.70000	0.00000	4791
1681	TO 4	BS	88.90000	0.00000	4792
1682	TO 5	BS	48.00000	0.00000	4793
1683	TO 6	BS	55.20000	0.00000	4794
1684	TO 7	BS	76.20000	0.00000	4795
1685	TO 8	BS	102.40000	0.00000	4796
1686	TO 9	BS	152.60000	0.00000	4797
1687	TO 10	BS	160.00000	0.00000	4798
1688	TO 11	BS	48.00000	0.00000	4799
1689	TO 12	BS	55.20000	0.00000	4800
1690	TO 13	BS	76.20000	0.00000	4801
1691	TO 14	BS	102.40000	0.00000	4802
.	.	.	.		
.	.	.	.		
.	.	.	.		
2044	TO367 D	BS	0.00000	0.00000	5155
2045	TO368	BS	12.20000	0.00000	5156
2046	TO369	BS	20.60000	0.00000	5157
2047	TO370	BS	30.10000	0.00000	5158
2048	TO371	BS	120.00000	0.00000	5159
2049	TO372	BS	131.40000	0.00000	5160
2050	TO373	BS	137.10000	0.00000	5161
2051	TO374	BS	145.20000	0.00000	5162
2052	TO375	BS	151.40000	0.00000	5163
2053	TO376	BS	162.20000	0.00000	5164
2054	TO377	BS	171.00000	0.00000	5165

Appendix N: Comparing TIS of LP Solution to Initial Schedule

This appendix contains an extract of the comparison of the time-in-system (TIS) for each piece of cargo in the three cargo flows ('TISCOMPx.DAT') as determined by a user-written program ('TISCOMP.FOR'). "DELTA" is the reduction in the TIS resulting from the LP solution's improvement of the initial flight schedule.

N.1 TIS Comparison for Cargo Flow #1

PIECE	SIZE	DELTA	PIECE	SIZE	DELTA	PIECE	SIZE	DELTA
1	3.60	4.80	204	5.00	1.20	407	5.00	14.40
2	3.60	4.80	205	0.67	1.20	408	1.19	5.80
3	3.60	7.20	206	5.00	1.20	409	4.92	4.80
4	3.59	7.20	207	0.67	1.20	410	2.25	4.80
5	3.60	9.60	208	5.00	1.20	411	1.12	4.80
6	3.60	9.60	209	1.69	12.00	412	3.01	12.50
7	3.60	8.80	210	1.70	2.40	413	5.00	12.50
8	1.76	0.00	211	1.45	4.80	414	3.43	38.40
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
37	5.00	26.30	240	1.20	24.80	443	3.74	13.80
38	5.00	77.30	241	1.20	24.80	444	3.75	13.80
39	5.00	77.30	242	1.20	24.80	445	3.75	13.80
40	5.00	77.30	243	1.20	24.80	446	1.51	14.40
41	5.00	77.30	244	1.20	24.80	447	1.50	18.80
42	5.00	77.30	245	1.20	24.80	448	1.51	18.80
43	1.17	7.20	246	1.20	24.80	449	1.63	4.80
44	5.00	7.20	247	1.56	24.80	450	1.62	4.80
45	5.00	7.20	248	0.52	0.80	451	1.63	38.40
46	5.00	7.20	249	0.52	0.80	452	0.41	1.90
47	5.00	7.20	250	0.73	9.00	453	0.41	1.80
48	5.00	7.20	251	1.38	31.20	454	1.80	9.60
49	5.00	7.20	252	1.68	7.20	455	1.80	5.80
50	5.00	7.20	253	1.68	7.20	456	1.81	5.80
51	1.17	9.10	254	1.68	8.80	457	1.80	9.00
52	5.00	9.10	255	1.68	8.80	458	1.80	9.00
53	5.00	9.10	256	0.22	30.10	459	1.80	9.00
54	5.00	9.10	257	0.18	86.40	460	2.65	5.30
55	5.00	9.10	258	5.00	0.00	461	2.66	5.30
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
201	0.67	7.20	404	5.00	16.10	607	1.13	1.80
202	5.00	7.20	405	5.00	14.40	608	1.13	1.80
203	0.67	1.20	406	4.88	14.40	609	1.13	1.80

MAX TIS IMPROVEMENT OF 86.40 HOURS MADE FOR PIECE # 257
MAX WEIGHTED-TIS IMPROVEMENT OF 386.50 TON-HOURS FOR PIECE # 38

N.2 TIS Comparison for Cargo Flow #2

PIECE	SIZE	DELTA	PIECE	SIZE	DELTA	PIECE	SIZE	DELTA
1	1.13	1.80	223	5.00	13.80	445	1.55	5.80
2	1.13	1.80	224	1.70	13.80	446	1.56	5.80
3	1.13	1.80	225	5.00	16.20	447	1.34	10.90
4	1.13	1.90	226	5.00	16.20	448	1.34	10.90
5	1.13	1.90	227	3.68	16.20	449	1.34	8.50
6	1.13	1.90	228	5.00	10.20	450	1.34	8.50
7	1.13	1.90	229	3.66	10.20	451	1.34	6.10
8	1.07	1.80	230	2.27	10.20	452	1.34	6.10
9	1.07	1.90	231	4.11	10.20	453	1.34	6.10
10	0.38	1.80	232	5.00	10.20	454	1.45	4.80
11	0.38	1.90	233	3.20	16.20	455	1.69	2.40
12	0.01	4.80	234	0.04	6.20	456	1.70	2.40
13	1.05	11.30	235	0.02	16.20	457	1.69	5.30
14	1.05	11.30	236	0.18	6.20	458	0.67	1.20
15	1.16	9.60	237	0.08	13.80	459	5.00	2.20
16	1.16	9.60	238	1.65	1.80	460	0.67	2.20
17	1.16	9.60	239	1.29	1.80	461	0.67	2.20
18	1.17	8.50	240	3.22	16.10	462	5.00	0.00
19	0.53	12.00	241	2.85	16.10	463	0.67	0.00
20	0.53	4.80	242	2.70	16.10	464	5.00	0.00
21	0.95	12.00	243	1.14	1.80	465	0.67	0.00
22	0.95	4.80	244	1.66	1.90	466	1.97	13.70
23	1.71	8.80	245	5.00	18.80	467	1.96	13.70
24	1.70	1.60	246	3.99	18.80	468	1.96	13.70
25	1.71	1.60	247	5.00	18.80	469	1.97	13.70
26	1.05	9.60	248	3.43	18.80	470	1.96	13.70
27	1.06	7.20	249	5.00	12.50	471	0.55	37.20
28	1.05	7.20	250	3.01	12.50	472	5.00	7.20
29	5.00	1.20	251	1.12	4.80	473	0.95	2.20
30	2.57	1.20	252	2.25	4.80	474	5.00	5.80
31	5.00	1.20	253	4.92	4.80	475	0.96	5.80
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
213	5.00	6.20	435	5.00	14.40	657	5.00	7.20
214	5.00	15.00	436	1.54	86.20	658	1.17	2.30
215	0.80	6.20	437	5.00	86.20	659	1.76	6.20
216	5.00	15.00	438	1.54	86.20	660	1.76	6.20
217	5.00	15.00	439	5.00	0.00	661	1.76	6.20
218	1.17	6.20	440	1.55	0.00	662	3.60	9.60
219	5.00	13.80	441	5.00	0.00	663	3.59	9.60
220	2.08	13.80	442	1.54	0.00	664	3.60	9.60
221	1.86	13.80	443	1.56	7.20	665	3.60	9.60
222	3.36	13.80	444	1.56	4.80	666	3.60	9.60

MAX TIS IMPROVEMENT OF 86.20 HOURS MADE FOR PIECE # 436
MAX WEIGHTED-TIS IMPROVEMENT OF 431.00 TON-HOURS FOR PIECE # 437

N.3 TIS Comparison for Cargo Flow #3

PIECE	SIZE	DELTA	PIECE	SIZE	DELTA	PIECE	SIZE	DELTA
1	3.60	6.10	208	5.00	43.80	415	3.26	34.90
2	1.17	33.60	209	3.31	9.60	416	1.18	9.60
3	5.00	33.60	210	3.26	13.10	417	5.00	9.60
4	5.00	33.60	211	1.30	24.00	418	5.00	9.60
5	5.00	33.60	212	1.19	60.00	419	5.00	9.60
6	5.00	33.60	213	5.00	60.00	420	0.67	1.20
7	5.00	33.60	214	5.00	9.60	421	5.00	1.20
8	5.00	33.60	215	5.00	9.60	422	1.34	13.10
9	5.00	33.60	216	0.72	20.10	423	1.55	38.00
10	1.28	11.30	217	1.68	60.00	424	5.00	14.20
11	4.46	11.30	218	1.37	15.80	425	1.20	20.10
12	5.00	11.30	219	0.89	8.80	426	3.70	20.10
13	1.65	7.20	220	0.96	5.80	427	5.00	20.10
14	1.48	28.80	221	5.00	9.60	428	5.00	20.10
15	0.81	2.40	222	1.96	15.80	429	5.00	20.10
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
65	2.12	8.90	272	1.26	1.90	479	2.40	1.40
66	3.[illegible]4	105.60	273	1.16	13.10	480	1.94	1.60
67	5.00	105.60	274	0.02	6.10	481	2.56	1.40
68	1.19	5.80	275	0.06	5.80	482	5.00	1.40
69	4.92	105.60	276	0.81	1.60	483	1.13	1.80
70	1.66	1.90	277	3.13	7.20	484	3.60	8.80
71	3.20	10.20	278	0.42	40.50	485	1.17	12.70
72	5.00	10.20	279	3.37	5.30	486	5.00	9.10
73	1.70	33.60	280	2.39	1.40	487	5.00	9.10
74	5.00	11.20	281	1.95	1.60	488	5.00	9.10
75	3.71	105.60	282	2.57	1.40	489	5.00	9.10
76	3.75	8.90	283	5.00	1.40	490	5.00	9.10
77	1.80	6.10	284	0.95	4.80	491	5.00	9.10
78	2.65	6.10	285	0.53	4.80	492	5.00	9.10
79	3.19	0.00	286	1.05	11.30	493	0.81	14.40
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
.	.	.	.	.	.	.		
200	5.00	24.00	407	5.00	9.10	614	4.28	1.80
201	1.65	7.20	408	1.29	48.80	615	1.85	42.80
202	1.47	40.50	409	1.48	13.90	616	1.16	8.80
203	1.48	40.50	410	0.81	14.40	617	3.38	14.40
204	1.21	4.80	411	5.00	14.40	618	2.39	1.20
205	0.80	21.60	412	0.16	9.60	619	2.57	1.20
206	5.00	21.60	413	5.00	9.60	620	5.00	1.20
207	0.16	43.80	414	3.32	9.60	621	1.13	1.80

MAX TIS IMPROVEMENT OF 105.60 HOURS MADE FOR PIECE # 75
MAX WEIGHTED-TIS IMPROVEMENT OF 528.00 TON-HOURS FOR PIECE # 67

Appendix O: DEMAND.FOR

This appendix contains the user-written FORTRAN program 'DEMAND.FOR'.

```
C
      PROGRAM DEMAND
C
C  This program will convert the current DEMAND.RAW file (or subset
C  of this file for the European Theater) into a file with "pieces"
C  of cargo. These "pieces" will be the "customers" to be transported
C  across the flight legs.
C
C  PAIRS = # OF ORIGIN-DESTINATION (O-D) PAIRS [W/ TRANSPORT DEMAND]
C  PIECES(*,?) = # OF 5-TON "PIECES" SHIPPED ON DAY ? FOR THAT O-D PAIR
C  SIZE(*,?) = SIZE (IN TONS) OF "SMALLER PIECES" (LESS THAN 5-TONS)
C           FOR THAT O-D PAIR (ROW *) SHIPPED ON DAY ?
C  OD(*,1) = ORIGIN BASE FOR ROW *
C  OD(*,2) = DESTINATION BASE FOR ROW *
C  CUMDEM(*,?) = CUMULATIVE DEMAND FOR WEEK AS OF DAY ? FOR ROW *
C  DAYDEM = DEMAND FOR ONE PARTICULAR DAY FOR WORKING O-D PAIR
C  MUSTGO = FLAG INDICATING NEED TO SHIP SMALL AMOUNT OF CARGO
C         BEFORE IT GETS ANY OLDER
C  COUNTER = COUNT OF THE NUMBER OF PIECES TO BE TRANSPORTED (THIS
C          REPRESENTS THE # OF "PIECE CONSTRAINTS" IN THE L.P.)
C
      INTEGER I, J, K, PAIRS, PIECES(464,7), COUNTER
      CHARACTER*4 OD(464,2)
      CHARACTER*1 MUSTGO
      REAL CUMDEM(464,7), SIZE(464,7), HEVDEM, MEDDEM, DAYDEM
      OPEN(UNIT=11,FILE='dmdeuro.dat',STATUS='OLD',IOSTAT=IERROR,
     &     ERR=911)
      COUNTER = 0
      DO 10 I = 1, 160
         READ(11,801, END=901) (OD(I,J), J=1,2), (CUMDEM(I,K), K=1,7)
   10 CONTINUE
  901 PAIRS = I-1
      CLOSE(11)
      OPEN(UNIT=12,FILE='dmdeuro.out',STATUS='UNKNOWN',IOSTAT=IERROR,
     &     ERR=912)
C Tracking the format of the output:
      WRITE(12,*) ' '
      WRITE(12,*) ' Format: O. & D. bases; (Cum Qty, Small piece size,
     & # of 5-ton pieces) * 7'
C NEED TO DETERMINE THE # OF PIECES OF CARGO
C FOR EACH O-D PAIR FOR EVERY WEEK... ASSUME THE FOLLOWING:
C   ... HEAVY DEMAND O-D PAIRS HAVE AT LEAST 5 TONS PER DAY AND SHIP
C        IN 5-TON "PIECES" PLUS THE REMAINDER
C   ... MEDIUM DEMAND O-D PAIRS HAVE AT LEAST 1 TON PER DAY AND
C        SHIP "PIECES" THE SIZE OF A SINGLE DAY'S DEMAND
```

```
C    ... LIGHT DEMAND O-D PAIRS HAVE LESS THAN 1 TON PER DAY, BUT SHIP
C        WHEN CUM. DEMAND REACHES > 1 TON OR (AT LEAST) EVERY 3 DAYS
      HEVDEM = 5.0
      MEDDEM = 1.0
      DO 50 I = 1, PAIRS
        PIECES(I,1) = 0
        SIZE(I,1) = 0.
        DAYDEM = CUMDEM(I,1)
        IF (DAYDEM.GE.HEVDEM) THEN
            PIECES(I,1) = INT(DAYDEM/HEVDEM)
            SIZE(I,1) = MOD(DAYDEM,HEVDEM)
          ELSE
            IF (DAYDEM.GE.MEDDEM) THEN
                SIZE(I,1) = DAYDEM
            ENDIF
        ENDIF
        IF (SIZE(I,1) .GT. 0.) THEN
            COUNTER = COUNTER + PIECES(I,1) + 1
          ELSE
            COUNTER = COUNTER + PIECES(I,1)
        ENDIF
        DO 40 J = 2, 7
          PIECES(I,J) = 0
          SIZE(I,J) = 0.
          MUSTGO = 'N'
          IF ((PIECES(I,J-1) .EQ. 0).AND.(SIZE(I,J-1) .eq. 0.)) THEN
              IF (J .EQ. 2) THEN
                  DAYDEM = CUMDEM(I,J)
                  GO TO 35
              ENDIF
              IF ((PIECES(I,J-2) .EQ. 0).AND.(SIZE(I,J-2) .EQ. 0.)) THEN
                  MUSTGO = 'Y'
                  IF (J .EQ. 3) THEN
                      DAYDEM = CUMDEM(I,J)
                      GO TO 35
                  ENDIF
                  DAYDEM = CUMDEM(I,J)-CUMDEM(I,J-3)
                ELSE
                  DAYDEM = CUMDEM(I,J)-CUMDEM(I,J-2)
              ENDIF
            ELSE
              DAYDEM = CUMDEM(I,J)-CUMDEM(I,J-1)
          ENDIF
   35     IF (DAYDEM.GE.HEVDEM) THEN
              PIECES(I,J) = INT(DAYDEM/HEVDEM)
              SIZE(I,J) = MOD(DAYDEM,HEVDEM)
            ELSE
              IF ((DAYDEM .GE. MEDDEM).OR.(MUSTGO .EQ. 'Y')) THEN
                  SIZE(I,J) = DAYDEM
              ENDIF
```

```
      ENDIF
      IF (SIZE(I,J) .GT. 0.) THEN
         COUNTER = COUNTER + PIECES(I,J) + 1
       ELSE
         COUNTER = COUNTER + PIECES(I,J)
      ENDIF
 40   CONTINUE
     WRITE(12,820) (OD(I,K),K=1,2),
   &            (CUMDEM(I,J), SIZE(I,J), PIECES(I,J), J=1,7)
 50 CONTINUE
     WRITE(12,*) 'TOTAL # OF PIECES NEEDING TRANSPORT =',COUNTER
 912 CLOSE(12)
    GO TO 1000
 801 FORMAT(A4,1X,A4,7(1X,F6.2))
 802 FORMAT(1X,A4,1X,A4,7(1X,F6.2))
 820 FORMAT(1X,A4,1X,A4,4(2(1X,F6.2),1X,I2),/,10X,3(2(1X,F6.2),1X,I2))
 821 FORMAT(1X,A4,1X,A4,2(1X,F6.2),1X,I2)
 911 PRINT*, 'REACHED END OF FILE MARKER BEFORE READING ALL DATA!'
1000 STOP
    END
```

Appendix P: SCHEDULD.FOR

This appendix contains the user-written FORTRAN program 'SCHEDULD.FOR'.

```
C
      PROGRAM SCHEDULD
C
C     This program takes existing information from AMC and combines it
C     to form a single file containing all the pertinent data about all
C     of the flights in the European theater in one month.  The output
C     file contains one line for each distinct flight, with each line
C     containing the following: flt # (assigned by this program), rte #,
C     which occurrence of the route, # of bases on rte., A/C capacity,
C     and then flt. leg info [dep. base, dep. time, arr. time, &
C     arr. base (which also covers the next dep. base if rte continues)].
C
C  RTID = I.D. OF CURRENT ROUTE
C  RTBASES = # OF BASES ON CURRENT ROUTE
C  RTSTOP(*) = STOPPING CODE FOR BASE * ON CURRENT ROUTE
C  RTBASE(*) = ICAO CODE FOR BASE * ON CURRENT ROUTE
C  OCCUR = # OF TIMES THE CURRENT ROUTE IS FLOWN IN ONE WEEK
C  SCHID(*) = ROUTE ID FOR SCHEDULE *
C  SCHAC(*) = AIRCRAFT TYPE (e.g., DC08 or C005) FOR SCHEDULE *
C  SCHDEP(*) = ORIG. DEPARTURE TIME (in days) FOR SCHEDULE *
C  FLYO(*) = ORIGIN BASE OF MISSION LEG
C  FLYD(*) = DESTINATION BASE OF MISSION LEG
C  FLYTIME(*) = FLIGHT TIME BETWEEN ORIGIN AND DEST. BASES
C  AC(*) = AIRCRAFT TYPE FOR OCCURRENCE * OF CURRENT ROUTE
C  DEPART(*) = ORIG. DEP. TIME FOR OCCURRENCE * OF CURRENT ROUTE
C  DEPTIM(*) = LEG (*) DEP. TIME FOR THIS OCCURRENCE OF CURRENT RTE
C  ARRTIM(*) = LEG (*) ARR. TIME FOR THIS OCCURRENCE OF CURRENT RTE
C  GRNTIM = LEG GROUND TIME FOR THIS OCCURANCE OF CURRENT ROUTE
C  FLTTIM = LEG FLIGHT TIME FOR THIS OCCURANCE OF CURRENT ROUTE
C    [note: all these times (____TIM) are in hours]
C  CAPAC = CAPACITY OF SPECIFIC AIRCRAFT FLYING GIVEN MISSION
C  NUMBAS = # OF BASES IN EUROPEAN THEATRE (not used)
C  NUMSCH = # OF SCHEDULED FLIGHTS FOR MONTH FOR ALL REGIONS
C  NUMFLY = # OF BASE COMBINATIONS (POSSIBLE LEGS) IN FILE FLY.DAT
C  NUMFLT = # OF DISTINCT FLIGHTS IN EUROPEAN THEATER IN 1 MONTH
C
      INTEGER I,J,K,L, RTBASES, RTID, RTSTOP(15), OCCUR
      INTEGER SCHID(612), CAPAC, NUMSCH, NUMFLY, NUMFLT
      REAL SCHDEP(612), FLYTIM(560), DEPART(20)
      REAL DEPTIM(15), FLTTIM, GRNTIM, ARRTIM(15), MULTIP
      CHARACTER*4 SCHAC(612), FLYO(560), FLYD(560), AC(20), RTBASE(15)
      OPEN(UNIT=13,FILE='rteeuro.dat',STATUS='OLD',ERR=93)
      OPEN(UNIT=14,FILE='schedule.raw',STATUS='OLD',ERR=94)
      OPEN(UNIT=15,FILE='fly.dat',STATUS='OLD',ERR=95)
      OPEN(UNIT=17,FILE='scheduld.prn',STATUS='UNKNOWN',ERR=97)
```

```
      WRITE(17,*) ' '
      WRITE(17,*) ' '

      DO 10 I = 1, 612
        READ(14,910,END=94) SCHID(I), SCHAC(I), SCHDEP(I)
   10 CONTINUE
   94 NUMSCH = I - 1
      CLOSE(14)

      DO 20 I = 1, 560
        READ(15,920,END=95) FLYO(I), FLYD(I), FLYTIM(I)
   20 CONTINUE
   95 NUMFLY = I - 1
      CLOSE(15)

      NUMFLT = 0

      DO 90 I = 1, 50
        RTBASES = 0

        DO 70 J = 1, 15
          RTSTOP(J) = 0
          RTBASE(J) = '    '
   70   CONTINUE

        READ(13,900,END=93) RTID, (RTBASE(J),RTSTOP(J), J=1,15)

        DO 80 J = 1, 15
          IF (RTSTOP(J) .GT. 0) RTBASES = RTBASES + 1
   80   CONTINUE
C         PRINT*, '# OF BASES ON ROUTE',RTID,'  IS',RTBASES

        OCCUR = 0
        DO 82 J = 1, NUMSCH
C             only include flights that begin before end of week
          IF ((RTID .EQ. SCHID(J)).AND.(SCHDEP(J) .LE. 7.0)) THEN
            OCCUR = OCCUR + 1
C             track occurence's scheduled dep. time & aircraft type
C             (convert dep. time from days into hours)
            DEPART(OCCUR) = SCHDEP(J) * 24.
            AC(OCCUR) = SCHAC(J)
          ENDIF
   82   CONTINUE

        CALL INSORT(OCCUR, DEPART, AC)

        DO 84 K = 1, OCCUR
          NUMFLT = NUMFLT + 1
          DEPTIM(1) = DEPART(K)
          FLTTIM = 0.
```

```
      DO 86 J = 1, RTBASES-1
       GRNTIM = 0.

       IF (RTSTOP(J) .EQ. 6) THEN
         IF (AC(K) .EQ. 'C005') GRNTIM = 18.25
         IF (AC(K) .EQ. 'C141') GRNTIM = 17.25
         IF (AC(K) .EQ. 'C130') GRNTIM = 16.25
         IF (AC(K) .EQ. 'DC08') GRNTIM = 16.00
         IF (AC(K) .EQ. 'DC10') GRNTIM = 16.00
         IF (AC(K) .EQ. 'B747') GRNTIM = 16.00
         IF (AC(K) .EQ. 'KC10') GRNTIM = 17.25
       ELSE
         IF (RTSTOP(J) .GT. 1) THEN
           IF (AC(K) .EQ. 'C005') GRNTIM = 4.25
           IF (AC(K) .EQ. 'C141') GRNTIM = 3.25
           IF (AC(K) .EQ. 'C130') GRNTIM = 2.25
           IF (AC(K) .EQ. 'DC08') GRNTIM = 3.00
           IF (AC(K) .EQ. 'DC10') GRNTIM = 4.00
           IF (AC(K) .EQ. 'B747') GRNTIM = 4.00
           IF (AC(K) .EQ. 'KC10') GRNTIM = 3.25
         ENDIF
       ENDIF

       IF (J .GT. 1) DEPTIM(J) = ARRTIM(J-1) + GRNTIM

       IF ((RTBASE(J) .EQ. 'EXXX').OR.(RTBASE(J) .EQ. 'KXXX').OR.
     &   (RTBASE(J+1) .EQ. 'EXXX').OR.(RTBASE(J+1) .EQ. 'KXXX')) THEN
         FLTTIM = 0.
       ELSE
         IF (AC(K) .EQ. 'C005') MULTIP = 0.97
         IF (AC(K) .EQ. 'C141') MULTIP = 1.00
         IF (AC(K) .EQ. 'C130') MULTIP = 1.39
         IF (AC(K) .EQ. 'DC08') MULTIP = 0.93
         IF (AC(K) .EQ. 'DC10') MULTIP = 0.92
         IF (AC(K) .EQ. 'B747') MULTIP = 0.91
         IF (AC(K) .EQ. 'KC10') MULTIP = 0.92
         DO 88 L = 1, NUMFLY
           IF ((RTBASE(J).EQ.FLYO(L)).AND.(RTBASE(J+1).EQ.FLYD(L)))
     &        FLTTIM = FLYTIM(L) * MULTIP
 88      CONTINUE
       ENDIF

       ARRTIM(J) = DEPTIM(J) + FLTTIM
 86   CONTINUE

      IF (AC(K) .EQ. 'C005') CAPAC = 50
      IF (AC(K) .EQ. 'C141') CAPAC = 18
      IF (AC(K) .EQ. 'C130') CAPAC = 7
      IF (AC(K) .EQ. 'DC08') CAPAC = 25
      IF (AC(K) .EQ. 'DC10') CAPAC = 40
```

```
      IF (AC(K) .EQ. 'B747') CAPAC = 71
      IF (AC(K) .EQ. 'KC10') CAPAC = 30

      WRITE(17,940) NUMFLT, RTID, K, RTBASES, CAPAC, (RTBASE(J),
     &        DEPTIM(J), ARRTIM(J), J=1,RTBASES-1), RTBASE(RTBASES)
 84   CONTINUE
 90 CONTINUE
 93 CLOSE(13)
 97 CLOSE(17)

900 FORMAT(I3, 15(1X,A4,I1))
910 FORMAT(I3,2X,A4,2X,F4.1)
920 FORMAT(2(A4,1X),6X,F4.1)
940 FORMAT(1X,2(I3,1X),3(I2,1X),4(/,18X,3(A4,1X,2(F5.1,1X))),
   &      /,18X,2(A4,1X,2(F5.1,1X)),A4)
    STOP
    END

    SUBROUTINE INSORT(N, DEPRAY, ACRAY)
C
C SUBROUTINE INSORT sorts N real values (dep. times) in a 1-dimensional
C array (DEPRAY) into ascending order by insertion-sort algorithm and
C also re-arranges the corresponding array (ACRAY) containing A/C type.
C
C Input argument: N =  the # of array elements to be sorted
C Two-way argument: DEPRAY = the Array (departure times) to be sorted
C              : ACRAY = Array to be kept in same order as DEPRAY
C Local constant: LIMIT = the size of the array
C
    INTEGER N, LIMIT
    PARAMETER (LIMIT = 30)
    REAL DEPRAY (1:LIMIT)
    CHARACTER*4 ACRAY (1:LIMIT)
C
C Internal variables: I & J are loop indices
C                 IMIN = Current position of minimum element
C                 MOVER = the minimum value in position IMIN
C                 XCHAR = the aircraft type in position IMIN
C
    INTEGER I, J, IMIN
    REAL MOVER
    CHARACTER*4 XCHAR
C
C Function invoked:
    REAL MINPOS
    EXTERNAL MINPOS
C Swap smallest element with first element:
    IMIN = MINPOS(N, DEPRAY)
    MOVER = DEPRAY(IMIN)
    DEPRAY(IMIN) = DEPRAY(1)
```

```
      DEPRAY(1) = MOVER
      XCHAR = ACRAY(IMIN)
      ACRAY(IMIN) = ACRAY(1)
      ACRAY(1) = XCHAR

C First and second elements are now sorted with respect to each other.
C Now move each of the remaining elements to its correct position in
C the array:
      DO 20 I = 3, N
         MOVER = DEPRAY(I)
         XCHAR = ACRAY(I)
         J=I
 10      IF (DEPRAY(J-1) .GT. MOVER) THEN
            DEPRAY(J) = DEPRAY(J-1)
            ACRAY(J) = ACRAY(J-1)
            J=J-1
            GO TO 10
         ENDIF
         DEPRAY(J) = MOVER
         ACRAY(J) = XCHAR
 20   CONTINUE
      END
C
C FUNCTION MINPOS:
C Finds subscript of DEPRAY element having lowest value.
C
      REAL FUNCTION MINPOS(N, DEPRAY)
C
C Input argument: N =  the # of array elements to be sorted
C Two-way argument: DEPRAY = the Array to be sorted
C Local constant: LIMIT = the size of the array
C
      INTEGER N, LIMIT
      PARAMETER (LIMIT = 30)
      REAL DEPRAY (1:LIMIT)
C
C Internal variables: I = loop index
C                     MINVAL = the currently-known minimum value
      INTEGER I
      REAL MINVAL
C
      MINVAL = DEPRAY(1)
      MINPOS = 1
      DO 50 I = 2, N
         IF (DEPRAY(I) .LT. MINVAL) THEN
            MINVAL = DEPRAY(I)
            MINPOS = I
         ENDIF
 50   CONTINUE
      END
```

Appendix Q: CARGFLOW.FOR

This appendix contains the user-written FORTRAN program 'CARGFLOW.FOR'.

```
C
      PROGRAM CARGFLOW
C
C    This program takes the cargo demand (cargo that needs to flow thru
C  the system), the schedule of flights, and the available transshipment
C  points to determine a feasible cargo flow.  Output details exactly
C  which legs of which flights are used to transport each piece of cargo
C  that the system can handle; output also details which pieces cannot
C  be flowed through the current system using this program.
C
C  Original version processed the cargo pieces (1 week of European
C  Theater) to go from #1 to #883; the second version processed the
C  cargo pieces in reverse order (from #883 to #1); and this third
C  version is designed to sort the Pieces of Cargo based on Mark-Times
C  so the Cargo Flow will be done in FIFO order.  Each version provides
C  a different set of sample data for the LP formulation.
C
C  UPDATE: 3/3/93
C  Changes:  corrected the output file 'carglegs.dat' to show the true
C          time-in-system (reflecting the 4 hours of assumed ground
C          time after cargo's last leg has landed) & made same fix to
C          the calculation of cum. Weighted Time-In-System
C
C  FLTNUM(*) = my designated flight number for this route & time comb.
C  RTID(*) = I.D. of route flown by FLTNUM *
C  OCCUR(*) = which occurance of this route
C  RTBASES(*) = # of bases on this route
C  CAPAC(*) = capacity of the aircraft flying this flight
C  RTBASE(*,?) = I.D. (4-letter ICAO) of base (?) on flight (*)
C  DEPTIM(*,?) = departure time for Leg (?) of flight (*)
C  ARRTIM(*,?) = arrival time for Leg (?) of flight (*)
C  LCAPAC(*,?) = available capacity on Leg (?) of flight (*)
C  OD(*,#) = "Origin-Destination" pair (*) base (#),
C            where 1 = Origin base  &   2 = Destination base
C  CUMDEM(*,?) = cumulative demand for O-D pair (*) as of day (?)
C  SIZE(*,?) = size of small (less than 5-ton) "piece" of cargo
C             generated by O-D pair (*) on day (?) of the week
C  PIECES(*,?) = # of big (5-ton) "pieces" of cargo generated by
C               O-D pair (*) on day (?) of the week
C  PIECE = count of total # of distinct cargo "pieces" for the week
C  CARGID(*,#) = Origin-Destination I.D. of cargo piece (*), where
C               # = 1 for Origin base  &  # = 2 for Destination base
C  CARGWT(*) = Weight of cargo piece (*)
C  CARGTM(*) = Time (day * 24 hours) of generation of cargo piece (*)
C  TRANSRTE(*,1) = Initial route used for transshipment route (*)
```

```
C TRANSRTE(*,#) = Follow-on route used for transshipment route (*),
C             where # can vary from 2 to 8
C FOLLOW(*) = # of follow-on routes for transshipment route (*)
C TRANBASE(*,?) = Base used for transshipment route (*),
C             where ? = 1 for Cargo's Origin Base,
C                   ? = 2 for Cargo's Transshipment Base,
C               and ? = 3 for Cargo's Destination Base
C LEGTOT = counter to accumulate the total # of legs flown in 1 week
C LEGNUM(*,#) = distinct LEG NUMber for the #th Leg of Flight (*)
C COUNT(*) = count of # of pieces transported by Leg (*)
C CARGO(*,?) = I.D. (piece #) of ?th piece transported by leg (*)
C MAXCNT = maximum # of pieces on any one Leg
C PIECECNT = counter used to track the pieces on each Leg
C LEG = counter used to iterate through each distinct Leg
C LEGID(*,?) = I.D. of LEG (*), giving Flt # (if ?=1) & which leg (?=2)
C PROCTIME(*) = PROCessing TIME of LEG (*)
C TOCONSTR = total # of CONSTRAINTS for Take-Off times (incl. non-neg.)
C RECONSTR = total # of CONSTRAINTS for Cargoes' Ready times
C FLOWED = counter used to track the # of pieces flowed thru system
C WTSYSTIM(*) = SYS TIM (Time in System) for piece (*) weighted by size
C           of the piece (units = tons * hours)
C CUMWTTIS = CUMulative of WTSYSTIM for all pieces flowed thru system
C WTFLOW(*) = WeighT (in tons) FLOWed through system for piece (*)
C CUMWTFLO = CUMulative of WTFLOW for all pieces flowed thru system
C CUMTOCON = CUMulative # of T.O. CONstraints prior to current leg
C KOUNT = counter used to iterate through each distinct cargo piece
C
      INTEGER I, J, K, L, FLTNUM(81), RTID(81), OCCUR(81), RTBASES(81)
      INTEGER CAPAC(81), PIECES(140,7), PIECE, CARGTM(884), LEGTOT
      INTEGER LEGNUM(81,15), TOTBAS, TRANSRTE(48,8), CARGO(377,20)
      INTEGER FOLLOW(48), M, N, P, R, S, T, V, W, COUNT(377), MAXCNT
      INTEGER LEGID(377,2), PIECECNT, LEG, TOCONSTR, FLOWED, RECONSTR
      CHARACTER*4 RTBASE(81,15),OD(140,2),CARGID(884,2),TRANBASE(48,3)
      REAL DEPTIM(81,15), ARRTIM(81,15), LCAPAC(81,15), CUMDEM(140,7)
      REAL SIZE(140,7), CARGWT(884), PROCTIME(1220), WTSYSTIM(884)
      REAL CUMWTTIS, CUMWTFLO, WTFLOW(884)
CCC   INTEGER CUMTOCON, KOUNT  (KOUNT is used for Flow #2 only)
      INTEGER CUMTOCON

      OPEN(UNIT=11,FILE='scheduld.prn',STATUS='OLD',ERR=91)

C Read in all scheduled flights available to transport cargo:
      READ(11,*)
      READ(11,*)
      LEGTOT = 0
      TOTBAS = 0
      CUMWTTIS = 0.
      CUMWTFLO = 0.
      DO 20 I = 1, 81
        DO 5 J = 1, 15
```

```
        RTBASE(I,J) = '   '
        DEPTIM(I,J) = 0.0
        ARRTIM(I,J) = 0.0
        PROCTIME((I-1)*15+J) = 0.0
        LCAPAC(I,J) = 0.0
        LEGNUM(I,J) = 0
 5    CONTINUE
      READ(11,910, END=10) FLTNUM(I), RTID(I), OCCUR(I),
     &                  RTBASES(I), CAPAC(I)
      READ(11,920, END=10) (RTBASE(I,J), DEPTIM(I,J),
     &                  ARRTIM(I,J), J=1,3)
      IF (RTBASES(I) .GT. 3) THEN
        READ(11,920, END=10) (RTBASE(I,J), DEPTIM(I,J),
     &                  ARRTIM(I,J), J=4,6)
        IF (RTBASES(I) .GT. 6) THEN
          READ(11,920, END=10) (RTBASE(I,J), DEPTIM(I,J),
     &                  ARRTIM(I,J), J=7,9)
          IF (RTBASES(I) .GT. 9) THEN
            READ(11,920, END=10) (RTBASE(I,J), DEPTIM(I,J),
     &                  ARRTIM(I,J), J=10,12)
            IF (RTBASES(I) .GT. 12) THEN
              READ(11,920, END=10) (RTBASE(I,J), DEPTIM(I,J),
     &                  ARRTIM(I,J), J=13,15)
            ENDIF
          ENDIF
        ENDIF
      ENDIF
 10   DO 15 J = 1, RTBASES(I)-1
        LCAPAC(I,J) = CAPAC(I)
        LEGTOT = LEGTOT + 1
        LEGNUM(I,J) = LEGTOT
        LEGID(LEGTOT,1) = FLTNUM(I)
        LEGID(LEGTOT,2) = J
        IF (J .LT. (RTBASES(I)-1)) THEN
          PROCTIME(LEGTOT) = DEPTIM(I,J+1) - DEPTIM(I,J)
         ELSE
          PROCTIME(LEGTOT) = ARRTIM(I,J) - DEPTIM(I,J) + 4
        ENDIF
 15   CONTINUE
      LCAPAC(I,RTBASES(I)) = CAPAC(I)
      TOTBAS = TOTBAS + RTBASES(I)
c       PRINT*, FLTNUM(I), '  ',(LEGNUM(I,J), J=1,RTBASES(I)-1)
 20 CONTINUE
 91 CLOSE(11)

c     PRINT*, 'TOTAL # OF FLT LEGS IN PROBLEM = ',TOTBAS-81

C   Read in all cargo demand (cargo requiring transport) and
C    place the distinct pieces into an array:
      OPEN(UNIT=12,FILE='dmdeuro.out',STATUS='OLD',ERR=92)
```

```
      READ(12,*)
      READ(12,*)
      PIECE = 0
      DO 40 I = 1, 140
        READ(12,940,END=40) (OD(I,K),K=1,2), (CUMDEM(I,J),
     &                 SIZE(I,J), PIECES(I,J), J=1,7)
        DO 30 J = 1, 7
          IF (SIZE(I,J) .GT. 0.) THEN
            PIECE = PIECE + 1
            CARGWT(PIECE) = SIZE(I,J)
            CARGID(PIECE,1) = OD(I,1)
            CARGID(PIECE,2) = OD(I,2)
            CARGTM(PIECE) = (J-1) * 24
          ENDIF
          IF (PIECES(I,J) .GT. 0) THEN
            DO 25 K = 1, PIECES(I,J)
              PIECE = PIECE + 1
              CARGWT(PIECE) = 5.0
              CARGID(PIECE,1) = OD(I,1)
              CARGID(PIECE,2) = OD(I,2)
              CARGTM(PIECE) = (J-1) * 24
   25       CONTINUE
          ENDIF
   30   CONTINUE
   40 CONTINUE
   92 CLOSE(12)

c     initialize the counter for # of pieces on each flight
      DO 45 I = 1, LEGTOT
        COUNT(I) = 0
        DO 43 J = 1, 15
          CARGO(I,J) = 0
   43   CONTINUE
   45 CONTINUE
      MAXCNT = 0
      FLOWED = 0
      TOCONSTR = 0
      RECONSTR = 0

c     Sort the Cargo Pieces by their Mark-Times so the Flow will be
c     completed in FIFO order. (& maintain each piece's weight & ID)
c [omit this next command for versions 1 & 2]
      CALL INSORT(PIECE, CARGTM, CARGID, CARGWT)

c     Write a list of all available Pieces of Cargo for the flow
c     OPEN(UNIT=16,FILE='cargpics.out',STATUS='UNKNOWN',ERR=96)
c     DO 50 I = 1, PIECE
c       WRITE(16,901) I,CARGID(I,1),CARGID(I,2),CARGTM(I),CARGWT(I)
c  50 CONTINUE
c  96 CLOSE(16)
```

```
c 901 FORMAT(1X,I3,2(2X,A4),2X,I3,2X,F4.2)

C  Read the Transshipment Data of available Transshipment Points
      OPEN(UNIT=13,FILE='trnseuro.dat',STATUS='OLD',ERR=93)
      DO 56 I = 1, 48
        FOLLOW(I) = 0
        DO 52 K = 1, 8
          TRANSRTE(I,K) = 0
   52   CONTINUE
        DO 53 J = 1, 3
          TRANBASE(I,J) = '   '
   53   CONTINUE
        READ(13,950,END=54) TRANSRTE(I,1), (TRANBASE(I,J),J=1,3),
     &      (TRANSRTE(I,K),K=2,8)
   54   DO 55 K = 2, 8
          IF (TRANSRTE(I,K) .GT. 0) FOLLOW(I) = FOLLOW(I) + 1
   55   CONTINUE
c       WRITE(*,950). TRANSRTE(I,1), (TRANBASE(I,J),J=1,3),
c    &            (TRANSRTE(I,K),K=2,FOLLOW(I)+1)
   56 CONTINUE
   93 CLOSE(13)


C  Find a Flight that can transport each piece of cargo. If no
C  Flight will work, check the list of transshipment points to
C  see if a combination of 2 routes is needed, and then check to
C  see if both of these routes have available capacity. If the
C  cargo piece cannot be transported using this logic, report it
C  as Unworkable; otherwise, report which legs of which Flights
C  were used to transport each piece.

      OPEN(UNIT=17,FILE='carglegs.dat3',STATUS='UNKNOWN',ERR=97)
      OPEN(UNIT=18,FILE='unflowed.dat3',STATUS='UNKNOWN',ERR=98)
      WRITE(17,*) ' '
      WRITE(17,*) ' '
      WRITE(17,965)
      WRITE(17,968)
      DO 175 K = 1, PIECE
c [replace the previous line with the following 2 lines for Version 2]
ccc    DO 175 KOUNT = 1, PIECE
ccc      K = PIECE+1 - KOUNT
        DO 160 I = 1, 81
          DO 150 J = 1, RTBASES(I)-1

c           need a route that includes the cargo's origin:
            IF (CARGID(K,1) .NE. RTBASE(I,J)) GO TO 150

c           need the Flight to depart after the cargo is generated:
            IF (REAL(CARGTM(K)) .GT. DEPTIM(I,J)) GO TO 150
```

```
c        need available capac. on this leg to handle cargo's weight:
         IF (CARGWT(K) .GT. LCAPAC(I,J)) GO TO 150

c           (I only get to here if leg's departure base, time,
c            & available capacity meet cargo's needs)

c        need the Flight to include the cargo's destination, and
c          need each leg of Flight to have available capac.:
         DO 65 L = J+1, RTBASES(I)
           IF (CARGWT(K) .GT. LCAPAC(I,L)) GO TO 150
           IF (CARGID(K,2) .NE. RTBASE(I,L)) GO TO 65

c          if I get to here, Orig, Dest, time, & capac. look good;
c          decrement the Capac of each leg and record the legs used:

           FLOWED = FLOWED + 1
           DO 60 M = J, L-1
             RECONSTR = RECONSTR + 1
             LCAPAC(I,M) = LCAPAC(I,M) - CARGWT(K)
             COUNT(LEGNUM(I,M)) = COUNT(LEGNUM(I,M)) + 1
             IF (MAXCNT .LT. COUNT(LEGNUM(I,M))) THEN
               MAXCNT = COUNT(LEGNUM(I,M))
             ENDIF
             CARGO(LEGNUM(I,M),COUNT(LEGNUM(I,M))) = FLOWED

c    output: new & old piece#, gen. time, weight, job#, leg used, plus
c    departure & arrival times and bases, time in system, & proc. time
           IF (M .EQ. (L-1)) THEN
             WRITE(17,902) FLOWED,K, CARGTM(K), CARGWT(K), M-J+1,
     &          LEGNUM(I,M), DEPTIM(I,M), ARRTIM(I,M),RTBASE(I,M),
     &          RTBASE(I,M+1), ARRTIM(I,M)-REAL(CARGTM(K)) + 4.,
     &          ARRTIM(I,M) - DEPTIM(I,M) + 4.0
             ELSE
             WRITE(17,902) FLOWED,K, CARGTM(K), CARGWT(K), M-J+1,
     &          LEGNUM(I,M), DEPTIM(I,M), ARRTIM(I,M),RTBASE(I,M),
     &          RTBASE(I,M+1), DEPTIM(I,M+1)-REAL(CARGTM(K)),
     &          DEPTIM(I,M+1) - DEPTIM(I,M)
             ENDIF
  60       CONTINUE
c            record the Weighted Time-in-System:
           WTSYSTIM(K) = (ARRTIM(I,L-1) + 4. -
     &                REAL(CARGTM(K))) * CARGWT(K)
c            record the Weight of piece flowed thru system:
           WTFLOW(K) = CARGWT(K)
c        -- mission complete for this piece of cargo, go to next piece
           GO TO 170

  65     CONTINUE

c      if I get to here, this Flight won't work without transshipment
```

```
c         ... so, check transshipment list ("trnseuro.dat"):
          DO 140 L = 1, 48
            IF (TRANBASE(L,1) .NE. CARGID(K,1)) GO TO 140
            IF (TRANBASE(L,3) .NE. CARGID(K,2)) GO TO 140

c           ...try to use this transshipment route:
            IF (RTID(I) .NE. TRANSRTE(L,1)) GO TO 140
            DO 120 N = 1, RTBASES(J)-1
              IF (TRANBASE(L,1) .NE. RTBASE(I,N)) GO TO 120
              IF (CARGWT(K) .GT. LCAPAC(I,N)) GO TO 120
              IF (REAL(CARGTM(K)) .GT. DEPTIM(I,N)) GO TO 120
              DO 110 P = N+1, RTBASES(I)
                IF (CARGWT(K) .GT. LCAPAC(I,P)) GO TO 120
                IF (TRANBASE(L,2) .NE. RTBASE(I,P)) GO TO 110

c               1st part of trans. route: N -- P is good!
c               check 2nd part (follow-on route):
                DO 100 R = 1, 81
                  DO 90 S = 2, FOLLOW(L)+1
                    IF (RTID(R) .NE. TRANSRTE(L,S)) GO TO 90

c         we get here if we have an available 1st part of transshipmt,
c           AND if we located a possible follow-on route (check feas.)
                    DO 80 T = 1, RTBASES(R)-1
                      IF (DEPTIM(R,T) .LT. (ARRTIM(I,P-1)+4)) GO TO 80
                      IF (TRANBASE(L,2) .NE. RTBASE(R,T)) GO TO 80
                      IF (CARGWT(K) .GT. LCAPAC(R,T)) GO TO 80
                      IF (REAL(CARGTM(K)) .GT. DEPTIM(R,T)) GO TO 80
                      DO 77 V = T+1, RTBASES(R)
                        IF (CARGWT(K) .GT. LCAPAC(R,V)) GO TO 80
                        IF (TRANBASE(L,3) .NE. RTBASE(R,V)) GO TO 77

c we only get here if we have available both parts of transshipmt rte.
c so, decrement the Capac of each leg used and record the legs used:
                        FLOWED = FLOWED + 1
                        DO 72 W = N, P-1
                          RECONSTR = RECONSTR + 1
                          LCAPAC(I,W) = LCAPAC(I,W) - CARGWT(K)
                          COUNT(LEGNUM(I,W)) = COUNT(LEGNUM(I,W)) + 1
                          IF (MAXCNT .LT. COUNT(LEGNUM(I,W))) THEN
                            MAXCNT = COUNT(LEGNUM(I,W))
                          ENDIF
                        CARGO(LEGNUM(I,W),COUNT(LEGNUM(I,W))) = FLOWED

c   output: new & old piece#, gen. time, weight, job#, leg used, plus
c   departure & arrival times and bases, time in system, & proc. time
                          IF (W .EQ. (P-1)) THEN
                            WRITE(17,903) FLOWED, K, CARGTM(K),
     &                         CARGWT(K), (W-N+1), LEGNUM(I,W),
     &                         DEPTIM(I,W), ARRTIM(I,W),
```

```
&                      RTBASE(I,W), RTBASE(I,W+1),
&                      ARRTIM(I,W) - REAL(CARGTM(K)) + 4.,
&                      ARRTIM(I,W) - DEPTIM(I,W) + 4.
                 ELSE
                   WRITE(17,903) FLOWED, K, CARGTM(K),
&                      CARGWT(K), (W-N+1), LEGNUM(I,W),
&                      DEPTIM(I,W), ARRTIM(I,W),
&                      RTBASE(I,W), RTBASE(I,W+1),
&                      DEPTIM(I,W+1) - REAL(CARGTM(K)),
&                      DEPTIM(I,W+1) - DEPTIM(I,W)
                 ENDIF
72               CONTINUE

                DO 74 M = T, V-1
                  RECONSTR = RECONSTR + 1
                  LCAPAC(R,M) = LCAPAC(R,M) - CARGWT(K)
                  COUNT(LEGNUM(R,M)) = COUNT(LEGNUM(R,M)) + 1
                  IF (MAXCNT .LT. COUNT(LEGNUM(R,M))) THEN
                    MAXCNT = COUNT(LEGNUM(R,M))
                  ENDIF
                  CARGO(LEGNUM(R,M),COUNT(LEGNUM(R,M)))=FLOWED

c    output: new & old piece#, gen. time, weight, job#, leg used, plus
c    departure & arrival times and bases, time in system, & proc. time
                  IF (M .EQ. (V-1)) THEN
                    WRITE(17,903) FLOWED, K, CARGTM(K),
&                      CARGWT(K), M-T+1+(P-N), LEGNUM(R,M),
&                      DEPTIM(R,M), ARRTIM(R,M),
&                      RTBASE(R,M), RTBASE(R,M+1),
&                      ARRTIM(R,M) - REAL(CARGTM(K)) + 4.,
&                      ARRTIM(R,M) - DEPTIM(R,M) + 4.
                  ELSE
                    WRITE(17,903) FLOWED, K, CARGTM(K),
&                      CARGWT(K), M-T+1+(P-N), LEGNUM(R,M),
&                      DEPTIM(R,M), ARRTIM(R,M),
&                      RTBASE(R,M), RTBASE(R,M+1),
&                      DEPTIM(R,M+1) - REAL(CARGTM(K)),
&                      DEPTIM(R,M+1) - DEPTIM(R,M)
                  ENDIF
74               CONTINUE
c                 record the Weighted Time-in-System:
                 WTSYSTIM(K) = (ARRTIM(R,V-1) + 4. -
&                          REAL(CARGTM(K))) * CARGWT(K)
c               record the Weight of piece flowed thru system:
                 WTFLOW(K) = CARGWT(K)
c        -- mission complete for this piece of cargo, go to next piece
                 GO TO 170

77               CONTINUE
80              CONTINUE
```

```
 90                   CONTINUE
 100                CONTINUE
 110              CONTINUE
 120            CONTINUE
c          -- this route is not available for transshipment
 140        CONTINUE
c        -- NO transshipment route is available for use with this Rte.
 150      CONTINUE
 160   CONTINUE
c      -- NO route is available for this Cargo (piece NOT flowed):
       WRITE(18,905) K, CARGTM(K), CARGWT(K)
       WTSYSTIM(K) = 0.0
       WTFLOW(K) = 0.0
 170 CUMWTTIS = CUMWTTIS + WTSYSTIM(K)
     CUMWTFLO = CUMWTFLO + WTFLOW(K)
 175 CONTINUE
  97 CLOSE(17)
  98 CLOSE(18)

c      output for each LEG: LEG #, FLT #, which Leg of this Flt,
c      the # of pieces transported & the ID of each piece
     OPEN(UNIT=19,FILE='legcargo.dat3',STATUS='UNKNOWN',ERR=99)
     WRITE(19,*) ' '
     WRITE(19,*) ' '
     CUMTOCON = 0
     DO 190 LEG = 1, LEGTOT
       IF (LEG .GT. 1) CUMTOCON = CUMTOCON + COUNT(LEG - 1) + 1
       WRITE(19,980) LEG, (LEGID(LEG, J), J=1,2), PROCTIME(LEG),
   &         COUNT(LEG), CUMTOCON,
   &         (CARGO(LEG, PIECECNT), PIECECNT=1,COUNT(LEG))
       DO 180 PIECECNT=1, COUNT(LEG) + 1
         TOCONSTR = TOCONSTR + 1
 180   CONTINUE
 190 CONTINUE
     WRITE(19,*) ' '
     WRITE(19,985) FLOWED
     WRITE(19,*) ' '
     WRITE(19,987) CUMWTTIS
     WRITE(19,*) ' '
     WRITE(19,989) CUMWTTIS/REAL(FLOWED)
     WRITE(19,*) ' '
     WRITE(19,991) CUMWTFLO
     WRITE(19,*) ' '
     WRITE(19,992) CUMWTFLO/REAL(FLOWED)
     WRITE(19,*) ' '
     WRITE(19,993) (CUMWTTIS/REAL(FLOWED))/(CUMWTFLO/REAL(FLOWED))
     WRITE(19,*) ' '
     WRITE(19,994) MAXCNT
     WRITE(19,*) ' '
     WRITE(19,995) TOCONSTR
```

```
      WRITE(19,*) ' '
      WRITE(19,997) RECONSTR+FLOWED
      WRITE(19,*) ' '
      WRITE(19,998) RECONSTR+FLOWED+TOCONSTR
      WRITE(19,*) ' '
      WRITE(19,999) FLOWED+TOCONSTR
   99 CLOSE(19)

  902 FORMAT(X,3(I3,2X),F4.2,2X,I2,2X,I3,
     &       2(2X,F6.2),2(2X,A4),2(2X,F6.2))
  903 FORMAT(X,3(I3,2X),F4.2,2X,I2,2X,I3,
     &       2(2X,F6.2),2(2X,A4),2(2X,F6.2),2X,'T')
  905 FORMAT(1X,I3,X,I3,X,F4.2)
  910 FORMAT(1X,2(I3,1X),3(I2,1X))
  920 FORMAT(18X,3(A4,1X,2(F5.1,1X)))
  940 FORMAT(1X,A4,1X,A4,4(2(1X,F6.2),1X,I2),/,10X,3(2(1X,F6.2),1X,I2))
  950 FORMAT(1X,I3,3(1X,A4),3X,7(1X,I3),2(1X,3I))
  965 FORMAT('New  Old  Mark  Size  job ',
     &      'Leg  Dep.   Arr.  Dep.  Arr.  Time     Proc')
  968 FORMAT(' #    # Time  (Wt)   #  ',
     &      ' #    Time    Time   Base  Base  in Sys    Time')
  980 FORMAT(1X,I3,1X,I3,1X,I2,1X,F4.1,1X,I2,1X,I4,20(1X,I3))
  985 FORMAT(1X,'Number of Pieces Flowed thru System  = ',I4)
  987 FORMAT(1X,'Cum. Weighted Time-in-System of pieces = ',F10.2)
  989 FORMAT(1X,'Avg. Weighted Time-in-System per piece = ',F10.2)
  991 FORMAT(1X,'Cum. Weight of Pieces Flowed thru System  = ',F10.2)
  992 FORMAT(1X,'Avg. Weight of Pieces Flowed thru System  = ',F10.2)
  993 FORMAT(1X,'Avg. Time-in-System of Pieces thru System  = ',F10.2)
  994 FORMAT(1X,'Max # of pieces on any one LEG = ',I2)
  995 FORMAT(1X,'# of constraints for T.O. Times (incl. non-neg.)= ',I4)
  997 FORMAT(1X,'# of constraints for Cargo Ready Times = ',I4)
  998 FORMAT(1X,'Total # of constraints in LP Formulation = ',I4)
  999 FORMAT(1X,'Total # of variables in LP Formulation = ',I4)

      STOP
      END


      SUBROUTINE INSORT(N, MTRAY, IDRAY, WTRAY)
C
C SUBROUTINE INSORT sorts N real values (mark times) in a 1-dimensional
C array (MTRAY) into ascending order by insertion-sort algorithm and
C also re-arranges the corresponding arrays (IDRAY & WTRAY) containing
C ID & Weight info.
C
C Input argument: N =  the # of array elements to be sorted
C Two-way argument: MTRAY = the Array (mark times) to be sorted
C                 : IDRAY = Array to be kept in same order as MTRAY
C Local constant: LIMIT = the size of the array (# of cargo pieces)
C
```

```
      INTEGER N, LIMIT
      PARAMETER (LIMIT = 884)
      INTEGER MTRAY (1:LIMIT)
      REAL WTRAY (1:LIMIT)
      CHARACTER*4 IDRAY (1:LIMIT,1:2)
C
C Internal variables: I & J are loop indices
C                  IMIN = Current position of minimum element
C                  MOVER = the minimum value in position IMIN
C                  XCHAR1 = the ID1 info in position IMIN
C                  XCHAR2 = the ID2 info in position IMIN
C                  XWT = the Weight of piece in position IMIN
C
      INTEGER I, J, IMIN, MOVER
      REAL XWT
      CHARACTER*4 XCHAR1, XCHAR2
C
C Function invoked:
      INTEGER MINPOS
      EXTERNAL MINPOS

C Swap smallest element with first element:
      IMIN = MINPOS(N, MTRAY)

      MOVER = MTRAY(IMIN)
      MTRAY(IMIN) = MTRAY(1)
      MTRAY(1) = MOVER

      XCHAR1 = IDRAY(IMIN,1)
      IDRAY(IMIN,1) = IDRAY(1,1)
      IDRAY(1,1) = XCHAR1

      XCHAR2 = IDRAY(IMIN,2)
      IDRAY(IMIN,2) = IDRAY(1,2)
      IDRAY(1,2) = XCHAR2

      XWT = WTRAY(IMIN)
      WTRAY(IMIN) = WTRAY(1)
      WTRAY(1) = XWT

C First and second elements are now sorted with respect to each other.
C Now move each of the remaining elements to its correct position in
C the array:

      DO 20 I = 3, N

         MOVER = MTRAY(I)
         XCHAR1 = IDRAY(I,1)
         XCHAR2 = IDRAY(I,2)
         XWT = WTRAY(I)
```

```
      J=I

10     IF (MTRAY(J-1) .GT. MOVER) THEN
          MTRAY(J) = MTRAY(J-1)
          IDRAY(J,1) = IDRAY(J-1,1)
          IDRAY(J,2) = IDRAY(J-1,2)
          WTRAY(J) = WTRAY(J-1)
          J=J-1
          GO TO 10
       ENDIF

       MTRAY(J) = MOVER
       IDRAY(J,1) = XCHAR1
       IDRAY(J,2) = XCHAR2
       WTRAY(J) = XWT

 20 CONTINUE
    END


C
C FUNCTION MINPOS:
C Finds subscript of MTRAY element having lowest value.
C
    INTEGER FUNCTION MINPOS(N, MTRAY)
C
C Input argument: N =  the # of array elements to be sorted
C Two-way argument: MTRAY = the Array to be sorted
C Local constant: LIMIT = the size of the array
C
    INTEGER N, LIMIT
    PARAMETER (LIMIT = 884)
    INTEGER MTRAY (1:LIMIT)
C
C Internal variables: I = loop index
C                     MINVAL = the currently-known minimum value
C
    INTEGER I, MINVAL
C
    MINVAL = MTRAY(1)
    MINPOS = 1
    DO 50 I = 2, N
       IF (MTRAY(I) .LT. MINVAL) THEN
          MINVAL = MTRAY(I)
          MINPOS = I
       ENDIF
 50 CONTINUE
    END
```

Appendix R: SCHEDMPS.FOR

This appendix contains the user-written FORTRAN program 'SCHEDMPS.FOR'.

```
C
      PROGRAM SCHEDMPS
C
C   This program takes the info on the flow of cargo (& legs used) and
C  transforms the data into the MPS format for my LP formulation of the
C  scheduling of AMC channel cargo missions in the European Theater.
C
C  RECONSTR = # of cargo REady-Time CONSTRaints
C  PIECENBR(*) = I.D. (PIECE # of cargo) for Cargo-Flow item (*)
C  JOBNBR(*) = JOB # (which job of cargo) for Cargo-Flow item (*)
C  LEGNBR(*) = LEG # used for transport of cargo for Cargo-Flow item (*)
C  MARKTIME(*) = MARK TIME (time created) for Cargo-Flow item (*)
C  SIZE(*) = SIZE of cargo for Cargo-Flow item (*)
C  PROCTIME(*) = PROCessing TIME of cargo for Cargo-Flow item (*)
C  TRANS(*) = indicator of whether Cargo-Flow item (*) gets TRANSshipped
C  RHS(*) = Right Hand Side of Constraint (*)
C  LEG(*) = LEG # of leg (*)
C  LEGID(*,?) = I.D. of LEG (*), giving Flt # (if ?=1) & which leg (?=2)
C  LEGCOUNT(*) = # of cargo pieces transported by LEG (*)
C  LEGCONSTR(*) = # of Take-Off Time constraints written for leg (*)
C  CARGO(*,?) = I.D. (piece #) of (?)th cargo transported by leg (*)
C  PIECECNT = counter to step through the pieces transported by leg (*)
C  LEGTOT = TOTal # of LEGs in this system
C  TOCONSTR = # of Take-Off Time Constraints (inluding non-negativity)
C  CUMTOCON(*) = CUMulative # of Take-Off CONstraints prior to leg (*)
C  LEGPROC(*) = PROCessing time (Flt time & Ground time) of LEG (*)
C  STOREROW(*,?) = STOREs the ROW #s of the Ready-Time Constraints that
C              need the Take-Off Time variable of leg (*) -- TOi
C  COUNTROW(*) = tally of the # of ROWs requiring  TOi
C
      INTEGER I, PIECENBR(1250), JOBNBR(1250), LEGNBR(1250), RECONSTR
      INTEGER J, LEGCOUNT(377), LEGCONSTR(377), LEG(377), LEGID(377,2)
      INTEGER PIECECNT, LEGTOT, TOCONSTR, CARGO(377,20), CUMTOCON(377)
      INTEGER STOREROW(377,20), MARKTIME(1250), COUNTROW(377)
      REAL RHS(3510), SIZE(1250), PROCTIME(1250)
      REAL LEGPROC(377)
      CHARACTER*1 TRANS(1250)

C     initialize RHS to signal errors & set Leg Proc. Times to 0
      DO 10 I= 1, 3510
       RHS(I) = 9999.
   10 CONTINUE

      DO 20 I= 1, 377
        COUNTROW(I) = 0
```

```
      DO 15 J= 1, 15
        STOREROW(I,J) = 0
 15     CONTINUE
      LEGPROC(I) = 0.
      LEGCONSTR(I) = 0
 20 CONTINUE

    OPEN(UNIT=16,FILE='schedmps.out',STATUS='UNKNOWN',ERR=96)
    WRITE(16,801)

    OPEN(UNIT=11,FILE='carglegs.out',STATUS='OLD',ERR=91)
    READ (11,805)

c   read the cargo-flow data; and
c   write the Type & Name of Constraints corresponding to Ready Times
    DO 30 I= 1, 1250
      READ (11,810,END=930) PIECENBR(I), MARKTIME(I), SIZE(I),
   &      JOBNBR(I), LEGNBR(I), PROCTIME(I), TRANS(I)
C  (this is now done below)
C       IF (LEGPROC(LEGNBR(I)) .EQ. 0.) THEN
C         LEGPROC(LEGNBR(I)) = PROCTIME(I)
C       ENDIF

      WRITE(16,820) I

 30 CONTINUE
930 RECONSTR = I-1
 91 CLOSE(11)

c   write Name & Type of the Extra ready-time constraints that define
c   the time that pieces reach their destinations [t_j(m_j + 1),j]
    DO 35 I= 1, PIECENBR(RECONSTR)

      WRITE(16,820) I + RECONSTR

 35 CONTINUE

c   read the file that tracks which cargo is transported by each leg;
c   track the accumulation of Constraints prior to any leg; and
c   write the Type & Name of Constraints corresponding to TakeOff Times
    OPEN(UNIT=12,FILE='legcargo.out',STATUS='OLD',ERR=92)
    READ(12,825)
    TOCONSTR = 0
    DO 50 I = 1, 377
      READ(12,830,END=950) LEG(I), (LEGID(I,J), J=1,2), LEGPROC(I),
   &        LEGCOUNT(I), (CARGO(I,PIECECNT), PIECECNT=1,LEGCOUNT(I))
      CUMTOCON(I) = TOCONSTR
c       WRITE(*,*) LEG(I), (LEGID(I,J), J=1,2),LEGPROC(I),LEGCOUNT(I),
c    &        (CARGO(I,PIECECNT), PIECECNT=1,LEGCOUNT(I)), CUMTOCON(I)
      DO 40 PIECECNT=1, LEGCOUNT(I) + 1
```

```
        TOCONSTR = TOCONSTR + 1

        WRITE(16,840) RECONSTR + PIECENBR(RECONSTR) + TOCONSTR

        RHS(RECONSTR + PIECENBR(RECONSTR) + TOCONSTR) = 0.
 40     CONTINUE
 50 CONTINUE
950 LEGTOT = I-1
c   PRINT*,'VALUES OF LEGTOT & TOCONSTR ARE',LEGTOT,' &',TOCONSTR
 92 CLOSE(12)


c
c       begin writing the Columns of the formulation
c       (for any variable, write the Name, Constraint, & Coefficient)
c
      WRITE(16,845)
c  set RHS to cargo's Mark Time for cargo's 1st leg, to Proc. Time
c  of previous leg for others, and to Proc. Time of this leg for last;
c  write the variables relating to Ready Times of the cargo [t__j(r),j]
c  (and store the locations of the rows that need TO variables)

      DO 60 I = 1, RECONSTR

c       determine the proper RHS for the current job of this piece
        IF (I .EQ. 1) THEN
          RHS(I) = REAL(MARKTIME(I))
        ELSE
          IF (JOBNBR(I) .EQ. 1) THEN
            RHS(I + PIECENBR(I-1)) = REAL(MARKTIME(I))
          ELSE
            RHS(I + PIECENBR(I) - 1) = PROCTIME(I-1)
          ENDIF
        ENDIF
c       determine the RHS for the End of the last job of this piece
        IF (I .EQ. RECONSTR) THEN
          RHS(I + PIECENBR(I)) = PROCTIME(I)
        ELSE
          IF (JOBNBR(I+1) .EQ. 1) THEN
            RHS(I + PIECENBR(I)) = PROCTIME(I)
          ENDIF
        ENDIF

c       write the Ready-Time constraint for this job
        IF (I .EQ. 1) THEN
          WRITE(16,850) LEGNBR(I), PIECENBR(I),'1 , 1.0
        ELSE
          IF (JOBNBR(I) .EQ. 1) THEN
            WRITE(16,850) LEGNBR(I), PIECENBR(I), I + PIECENBR(I-1), 1.0
          ELSE
```

```
        WRITE(16,850) LEGNBR(I), PIECENBR(I), I + PIECENBR(I)-1, 1.0
       ENDIF
      ENDIF

      LEGCONSTR(LEGNBR(I)) = LEGCONSTR(LEGNBR(I)) + 1

      WRITE(16,850) LEGNBR(I), PIECENBR(I),
     &      RECONSTR + PIECENBR(RECONSTR) + CUMTOCON(LEGNBR(I)) +
     &      LEGCONSTR(LEGNBR(I)), -1.0

      IF (JOBNBR(I) .GE. 2) THEN
c       need the TO variable of the previous leg:
       COUNTROW(LEGNBR(I-1)) = COUNTROW(LEGNBR(I-1)) + 1
       STOREROW(LEGNBR(I-1),COUNTROW(LEGNBR(I-1))) = I+PIECENBR(I)-1
      ELSE
c       if (jobnbr(i) .eq. 1), write to Objective Function:
       WRITE(16,870) LEGNBR(I), PIECENBR(I), -SIZE(I)
      ENDIF

      IF (I .EQ. RECONSTR) THEN
       WRITE(16,855) PIECENBR(I), I + PIECENBR(I), 1.0
       COUNTROW(LEGNBR(I)) = COUNTROW(LEGNBR(I)) + 1
       STOREROW(LEGNBR(I),COUNTROW(LEGNBR(I))) = I + PIECENBR(I)
       WRITE(16,875) PIECENBR(I), SIZE(I)
      ELSE
       IF (JOBNBR(I+1) .EQ. 1) THEN
        WRITE(16,855) PIECENBR(I), I + PIECENBR(I), 1.0
        COUNTROW(LEGNBR(I)) = COUNTROW(LEGNBR(I)) + 1
        STOREROW(LEGNBR(I).COUNTROW(LEGNBR(I))) = I + PIECENBR(I)
        WRITE(16,875) PIECENBR(I), SIZE(I)
       ENDIF
      ENDIF
   60 CONTINUE


c     write the variables relating to the Take-Off Times of the Legs
     DO 90 I = 1, LEGTOT

      K = RECONSTR + PIECENBR(RECONSTR) + CUMTOCON(I)

      DO 70 J = K+1, K+LEGCOUNT(I)+1
       WRITE(16,860) I, J, 1.0
   70  CONTINUE

      IF (LEGID(I+1,2) .GE. 2) THEN
       WRITE(16,860) I, K + LEGCOUNT(I) + 1 + LEGCOUNT(I+1) + 1, -1.0
       RHS(K + LEGCOUNT(I) + 1 + LEGCOUNT(I+1) + 1) = LEGPROC(I)
      ENDIF

      DO 80 J = 1, COUNTROW(I)
```

```
      WRITE(16,860) I, STOREROW(I,J), -1.0
 80   CONTINUE

 90 CONTINUE


c       write the Right-Hand Sides (RHS) of every Constraint
     WRITE(16,885)
     DO 100 I = 1, RECONSTR + PIECENBR(RECONSTR) + TOCONSTR
       WRITE(16,890) I, RHS(I)
 100 CONTINUE

     WRITE(16,895)
 96 CLOSE(16)

 801 FORMAT('NAME SCHEDULING LP( MIN)',/,'ROWS',/,2X,'N',1X,'OBJ')
 805 FORMAT(///)
 810 FORMAT(X,I3,2X,3X,2X,I3,2X,F4.2,2X,I2,2X,I3,16X,22X,F6.2,2X,A1)
 820 FORMAT(2X,'E',1X,'R',I4)
 825 FORMAT(/)
 830 FORMAT(1X,I3,1X,I3,1X,I2,1X,F4.1,1X,I2,5X,20(1X,I3))
 840 FORMAT(2X,'G',1X,'R',I4)
 845 FORMAT('COLUMNS')
 850 FORMAT(4X, 't', I3, ',', I3, 2X, 'R', I4, 5X, F5.2)
 855 FORMAT(4X, 'tEND,', I3, 2X, 'R', I4, 5X, F5.2)
 860 FORMAT(4X, 'TO', I3, 5X, 'R', I4, 5X, F5.2)
 870 FORMAT(4X, 't', I3, ',', I3, 2X, 'OBJ', 7X, F5.2)
 875 FORMAT(4X, 'tEND,', I3, 2X, 'OBJ', 7X, F5.2)
 885 FORMAT('RHS')
 890 FORMAT(4X, 'RHS', 7X, 'R', I4, 5X, F5.1)
 895 FORMAT('ENDATA')

     STOP
     END
```

Appendix S: TISCOMP.FOR

This appendix contains the user-written FORTRAN program 'TISCOMP.FOR'.

```
C
      PROGRAM TISCOMPARE
C
C  This program will compare the Time-in-System results from the
C original cargo flow to those of the optimal solution from the LP.
C
C PIECEA = ID OF CARGO PIECE (ON ITS FIRST LEG)
C TISA = TIME-IN-SYSTEM (TIS) OF CARGO PIECE (ON ITS FIRST LEG)
C SIZEA = SIZE OF CARGO PIECE (ON ITS FIRST LEG)
C PIECEB = ID OF "NEXT" CARGO PIECE
C TISB = TIME-IN-SYSTEM OF "NEXT" CARGO PIECE
C SIZEB = SIZE OF "NEXT" CARGO PIECE
C OLDTIS(*) = TIME-IN-SYSTEM OF CARGO PIECE FROM ORIG. CARGO FLOW
C SIZE(*) = SIZE (IN TONS) OF THE (*) CARGO PIECE
C FLTLEGA = FLIGHT LEG OF CARGO PIECE (ON ITS FIRST LEG)
C MARKTIMA = MARK TIME OF CARGO PIECE (ON ITS FIRST LEG)
C FLTLEGB = FLIGHT LEG OF CARGO PIECE (ON ITS NEXT LEG)
C MARKTIMB = MARK TIME OF CARGO PIECE (ON ITS NEXT LEG)
C NEWTIS(*) = TIME-IN-SYSTEM OF CARGO PIECE FROM LP SOLUTION
C DELTA(*) = IMPROVEMENT IN TIS FROM ORIG. FLOW TO LP SOL.
C MAXID = ID OF PIECE WITH MAXIMUM VALUE OF DELTA
C MAXIDWT = ID OF PIECE WITH MAXIMUM VALUE OF WEIGHTED DELTA,
C          WHERE WEIGHTED DELTA = DELTA * SIZE OF PIECE
C PIECES = TOTAL NUMBER OF CARGO PIECES IN THIS COMPARISON
C
      INTEGER I, PIECEA, PIECEB, MAXID, MAXIDWT, PIECES
      CHARACTER*4 FLTLEGA, FLTLEGB
      REAL SIZE(700), TISA, TISB, OLDTIS(700), MARKTIMA, MARKTIMB
      REAL NEWTIS(700), DELTA(700), SIZEA, SIZEB
      MAXID = 1
      MAXIDWT = 1
      OPEN(UNIT=11,FILE='carglegs.dat1',STATUS='OLD')
      READ(11,805, END=901)
      I = 1
      READ(11,810, END=901) PIECEA, SIZEA, TISA
   50 READ(11,810, END=901) PIECEB, SIZEB, TISB
      IF (PIECEA .EQ. PIECEB) THEN
        TISA = TISB
        GO TO 50
      ENDIF
      OLDTIS(I) = TISA
      SIZE(I) = SIZEA
      PIECEA = PIECEB
      TISA = TISB
```

```
      SIZEA = SIZEB
      I = I + 1
      GO TO 50
  901 PIECES = I
      OLDTIS(I) = TISA
      SIZE(I) = SIZEA
      CLOSE(11)
  805 FORMAT(///)
  810 FORMAT(X,I3,12X,F4.2,39X,F6.2)
      OPEN(UNIT=12,FILE='schedrun1.sol',STATUS='OLD')
      READ(12,815, END=901)
      I = 1
   90 READ(12,820, END=902) FLTLEGA, MARKTIMA
  100 READ(12,820, END=902) FLTLEGB, MARKTIMB
      IF (FLTLEGB .NE. 'tEND') GO TO 100
      NEWTIS(I) = MARKTIMB - MARKTIMA
C       calculate the improvement of TIS for this piece:
      DELTA(I) = OLDTIS(I) - NEWTIS(I)
C       check if this improvement is largest so far:
      IF (DELTA(I) .GT. DELTA(MAXID)) THEN
        MAXID = I
      ENDIF
C       check if the weighted improvement is largest so far:
      IF ((DELTA(I)*SIZE(I)) .GT. DELTA(MAXIDWT)*SIZE(MAXIDWT)) THEN
        MAXIDWT = I
      ENDIF
      I = I + 1
      GO TO 90
  902 CLOSE(12)
  815 FORMAT(/////////////)
  820 FORMAT(10X,A4,17X,F6.2)
      OPEN(UNIT=16,  `.E='tiscomp1.dat',STATUS='UNKNOWN')
      WRITE (16,825)
      INCREMNT = PIECES/3
      DO 150 I = 1, INCREMNT
        WRITE (16,830) I, SIZE(I), DELTA(I),
     &          I+INCREMNT, SIZE(I+INCREMNT), DELTA(I+INCREMNT),
     &          I+INCREMNT*2,SIZE(I+INCREMNT*2),DELTA(I+INCREMNT*2)
  150 CONTINUE
      WRITE (16,840) DELTA(MAXID), MAXID,
     &           DELTA(MAXIDWT) * SIZE (MAXIDWT), MAXIDWT
      CLOSE(16)
  825 FORMAT(//,3(4X,'PIECE',2X,'SIZE',3X,'DELTA'))
  830 FORMAT(3(6X,I3,2X,F4.2,2X,F6.2))
  840 FORMAT(1X,'MAX TIS IMPROVEMENT OF ',F6.2,
     &        ' HOURS MADE FOR PIECE # ',I3,/
     &      1X,'MAX WEIGHTED-TIS IMPROVEMENT OF ',F6.2,
     &        ' TON-HOURS MADE FOR PIECE # ',I3)
 1000 STOP
      END
```

Appendix T: Estimate of Piece-Legs for Entire System

This appendix contains the totals for the number of legs flown by each type of aircraft. These totals, when multiplied by the average number of pallet positions used on each leg (based on the utilization rate from 'PLANES.OUT') and then summed, provide an estimate for the number of *piece-legs* for the entire AMC channel system for one month. This estimate is used to estimate the number of constraints required to model the scheduling of the system with the LP formulation.

	tot flts	tot legs	C-5 flts	C-5 legs	C141 flts	C141 legs	C130 flts	C130 legs
TOTALS:	607	2757		271		1773		319
ute rate:				0.552		0.565		0.561
pallet pos max:				34		12		5
avg # pos used:				18.76		6.782		2.804
(avg#pos * #legs):				5084		12025		894

	B747 flts	B747 legs	DC8 flts	DC8 legs	DC10 flts	DC10 legs	KC10 flts	KC10 legs
TOTALS:		169		136		9		80
ute rate:		0.741		0.611		0.526		0.729
pallet pos max:		48		16		26		20
avg # pos used:		35.56		9.776		13.67		14.57
(avg#pos * #legs):		6009		1330		123		1166

TOTALS for ALL:
flt's = 607
legs = 2757
legs/flt= 4.54
pic-leg 26632

Appendix U: Estimate of Piece-Legs for European Theater

This appendix contains the totals for the number of legs flown by each type of aircraft. These totals, when multiplied by the average number of pallet positions used on each leg (based on the utilization rate from 'PLANES.OUT') and then summed, provide an estimate for the number of *piece-legs* for the European theater for one month. This estimate is used to estimate the number of constraints required to model the scheduling of one month of this theater with the LP formulation.

	tot flts	tot legs	C-5 flts	C-5 legs	C141 flts	C141 legs	C130 flts	C130 legs
TOTALS:	261	1228		107		737		148
ute rate:				0.552		0.565		0.561
pallet pos max:				34		12		5
avg # pos used:				18.76		6.782		2.804
(avg#pos * #legs):				2007		4999		415

	B747 flts	B747 legs	DC8 flts	DC8 legs	DC10 flts	DC10 legs	KC10 flts	KC10 legs
TOTALS:		83		75		9		69
ute rate:		0.741		0.611		0.526		0.729
pallet pos max:		48		16		26		20
avg # pos used:		35.56		9.776		13.67		14.57
(avg#pos * #legs):		2951		733		123		1006

TOTALS for ALL:
flt's = 261
legs = 1228
legs/flt= 4.70
pic-leg 12234

Bibliography

Ackley, M., W. Carter, G. Hughes, J. Litko, K. Ware, A. Whisman, and R. Roehrkasse. *Optimization Applications at the Military Airlift Command: Importance and Difficulties*. Unpublished paper provided to the Second International Conference on Industrial and Applied Mathematics. Washington, D.C., July 8-12, 1991.

Ackley, M., B. Carter, J. Litko, K. Ware and A. Whisman. *Decision Support for the Air Mobility Command Scheduled Cargo System*. Unpublished paper provided by LTC J. Litko, Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL.

Baker, Kenneth R. *Introduction to Sequencing and Scheduling*. New York: John Wiley and Sons, 1974.

Berg, CAPT Wayne. Personal interview. Tanker and Airlift Control Center (TACC), Air Mobility Command, HQ AMC/TRKIA, Scott AFB, IL. 22 September 1992.

Bodin, Lawrence D. "Twenty Years of Routing and Scheduling," *Operations Research*, 38: 571-579 (July-August 1990).

Borsi, MAJ John J. "Possible MAC Thesis Topics". Notes for Students in GOR-93M. Air Force Institute of Technology, Wright-Patterson AFB, OH. 11 April 1992.

Carter, Brand and Joseph R. Litko. *Simulating the Air Mobility Command Channel Cargo System*. Unpublished paper provided by LTC J. Litko, Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL.

Del Rosario, CPT Michael. *Determining Cargo Flow for Air Mobility Command's Channel Cargo System*. MS thesis, AFIT/GOR/ENS/93M-04. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1993 (Note: No DTIC assigned yet).

French, Simon. *Sequencing and Scheduling*. New York: Ellis Horwood, 1982.

Litko, LTC Joseph R. Personal interview. Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL. 22 September 1992.

Litko, LTC Joseph R. Telephone interview. Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL. 26 August 1992.

Litko, LTC Joseph R. Telephone interview. Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL. 9 September 1992.

Litko, LTC Joseph R. Telephone interview. Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL. 13 October 1992.

Moul, Capt Justin E. *A Method for Determining Schedule Delay Information in a Channel Cargo Route Network Schedule*. MS thesis, AFIT/GST/ENS/92M-05 School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, June 1992 (Note: No DTIC assigned yet).

Nemhauser, George L. and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. New York: John Wiley & Sons, 1988.

Robinson, 1LT Jonathan. Personal interview and data derived from an AMC study. Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL. 22 September 1992.

Schrage, Linus. *User's Manual for LINDO*. Redwood City, CA: The Scientific Press, 1987.

Whisman, Alan. Correspondence. Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL. 27 October 1992.

Whisman, Alan. Personal interview. Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL. 22 September 1992.

Whisman, Alan. Telephone interview. Command Analysis Group, Air Mobility Command, HQ AMC/XPYR, Scott AFB, IL. 30 October 1992.

Vita

Captain Gregory S. Rau was born on 25 January 1965 in Bethlehem, Pennsylvania. He graduated from Southern Lehigh High School, in Center Valley, Pennsylvania, on 18 June 1982 and attended the United States Air Force Academy in Colorado Springs, Colorado. After graduating with a Bachelor of Science in Operations Research in May of 1986, he attended Undergraduate Pilot Training for several months before being re-assigned in February 1987 to the Air Force Logistics Management Center (AFLMC) at Gunter AFB, Alabama, as an Operations Research Analyst. Captain Rau worked as an analyst on a variety of projects pertaining to Supply, Maintenance, Contracting, and Transportation during his four and one-half years at the AFLMC. In August 1991, he was assigned to the Graduate Operations Research program in the School of Engineering at the Air Force Institute of Technology, Wright-Patterson AFB, Ohio. Following his March 1993 graduation, Captain Rau received an assignment as an instructor in the Math Department at the United States Air Force Academy. He and his wife, Kim, had their first child, Coleman Gregory, while at AFIT.

Permanent Address: 5128 West Saucon Avenue
Center Valley, PA 18034

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1993	Master's Thesis

4. TITLE AND SUBTITLE
SCHEDULING AIR MOBILITY COMMAND'S CHANNEL CARGO MISSIONS

5. FUNDING NUMBERS

6. AUTHOR(S)
Gregory S. Rau, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
WPAFB, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/93M-19

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
HQ AMC/XPYR
SCOTT AFB IL 62225

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Approved for public release; distribution unlimited

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Through the use of a linear programming model, this research revised the initial schedule for AMC's channel cargo missions to eliminate any excess delay enroute by minimizing the cumulative, weighted time-in-system for all cargo, according to a given cargo flow. In fact, the revised schedule minimizes any assigned nonnegative weighting of the time-in-system, due to the properties of equivalent measures of performance. When combined with Step One of a proposed two-step process for revising AMC's channel mission schedule, this research can be used to improve the current schedule based on Step One's cargo flow. By carefully defining the notation and adapting the job-shop formulation, this research devised a method for modeling the scheduling of a limited-size portion of AMC s channel system and minimizing the delay enroute. If future research can improve this method using the recommendations provided, this method could become a significant part of AMC's advance planning process.

14. SUBJECT TERMS
Scheduling, Logistics, Transportation, Transshipment, Linear Programming, Dual Variables

15. NUMBER OF PAGES
132

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

END

FILMED

DATE: 4-93

DTIC